NORDDESIGN CONFERENCE
REYKJAVIK, ICELAND, 12-14 AUGUST 2024

Testing the Generalizability of Deep Learning Based Plausibility Detection with Unknown Finite Element Simulations

Sebastian Bickel, Stefan Goetz, Sandro Wartzack

Friedrich-Alexander-Universität Erlangen-Nürnberg, Engineering Design, Germany

Abstract: Shorter development times and greater product variety are necessary in the current business world. These challenges can be managed with virtual testing methods, but these require experienced engineers. To counteract the shortage, a data-driven plausibility detection was developed. The plausibility is determined with Deep Learning models that are trained on existing simulations. The practical application requires the models to work with simulations not included in the training data. Therefore, this paper analyzes the transferability of a plausibility detection model to new, unknown instances.

Keywords: Artificial Intelligence (AI), Data Driven Design, Structural Analysis, Deep Learning, Finite Element Simulation

1 Introduction

Artificial Intelligence (AI) methods are an integral part in our daily lives, for example, when recognizing road signs, translating text in images with smartphones or answering questions with chatbots. Consequently, the application of such techniques in industry is considerably increasing, as recent studies from (Dumbach et al., 2023) confirm. In addition to data availability and method development, transferability between the training and application environment is another problem for the successful deployment of such new methods and models. A study of different sign recognition systems from various manufacturers (Ippen and Bach, 2019) revealed that recognition accuracies range between 30 % and 95 % and therefore deviate widely. This paper seeks to test the transferability of a method to assist users of Finite Element simulation to reduce the incidence of such problems. The method is a data-driven approach that utilizes existing simulation data to predict the plausibility of new simulations using Deep Learning. This is intended to primarily support inexperienced users and enable them to conduct simulations themselves. Consequently, the state of the art on plausibility detection and testing with unknown data is presented first, followed by the research gap and approach. Afterwards, the adaptation of the test procedure to FE simulations is presented and then applied. An evaluation and discussion of the results summarizes the findings in this paper, which finally closes with a conclusion and outlook.

2 State of the art and fundamentals

First the preexisting data-driven plausibility detection method for FE simulations is described, followed by the fundamental metrics for evaluating trained models and approaches to test their generalization ability for unknown data.

2.1. Plausibility detection

Several tools assisting the setup or evaluation of FE simulations were created in specific application domains, for example, the SACON system of (Bennett et al., 1978), which helped to develop the wings for a Boeing 747. Another example is the online support system from (Woyand et al., 2012), which uses a knowledge database to facilitate training in the FE environment of CATIA. Both of these methods use a knowledge database as the source for assistance, which is very time-consuming to create and maintain. In contrast, an approach to classify FE simulations according to their plausibility was developed by (Spruegel et al., 2015; Spruegel et al., 2021), which relies on the utilization of existing FE data. The term plausibility is defined by (Spruegel et al., 2015) as an FE simulation that contains no obvious errors and can be paraphrased by the term ‘likely valid’. Therefore, an experienced simulation engineer would recognize these errors, which include incorrectly assigned units (e.g., bar and MPa), missing bearings or excessively coarse meshing.

The procedure for checking the plausibility of FE simulations is shown in Figure 1. The start of the process is the FE simulation, whose structure and results are transferred to matrices through a projection method. For this purpose, the mesh is projected onto a sphere, which is divided into different sections. The values of the projected nodes are collected at these areas and then assigned to a matrix, similar to the transformation of a globe to a map. The matrices can subsequently serve as input for a Convolutional Neural Network (CNN) which, after successful training, can recognize the plausibility of new simulations. A detailed description of this process can be found in (Spruegel et al., 2021). A similar idea for the classification of Acoustic Finite Element Simulation with PointNet and DGCNN was also developed, but tested with a much smaller dataset (Shivaditya et al., 2022).

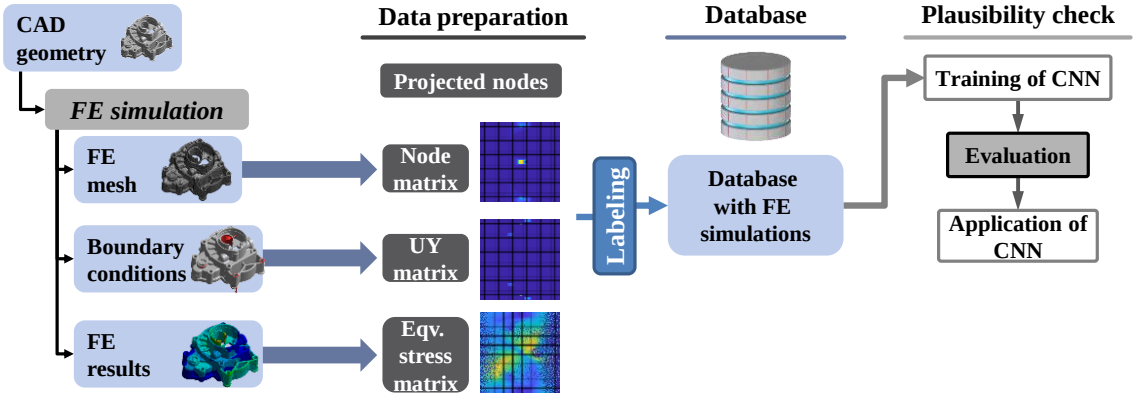

Figure 1. Overview of the plausibility-check method according to (Spruegel et al., 2015; Spruegel et al., 2021)

2.2. Testing models with unknown data

The evaluation of Machine or Deep Learning models is an essential step in the application of new algorithms or models. Therefore, it is also a key step in all procedures for employing trained models, for example in CRISP-DM or Knowledge Discovery in Databases (KDD). Numerous factors, such as the objective, the model, the data and the metrics, influence the evaluation result.

At first, the metrics relevant to the problem area must be determined, whereby a distinction is made between the general function of the model, e.g., whether it is a classification, regression or clustering problem. Since the presented procedure for the plausibility check is a classification problem, only metrics for the evaluation of classification tasks are presented. The different options for evaluating models with unknown data are then presented, starting with the classic hold-out approach, followed by model robustness and concluding with the out-of-sample technique.

2.2.1. Classification metrics

The basis for the analysis of classification problems is the confusion matrix. It consists of the variables True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The metric used to compare the different Deep Learning architectures is called accuracy and is normally calculated for the three datasets: the training, validation, and test datasets (Fawcett, 2006; Powers, 2011; Zhou, 2021). The comparison between training and test accuracy also provides helpful information about the model. It is useful to look at the difference between the two accuracies to determine overfitting or underfitting. Overfitting describes the phenomenon when the model is too well adapted to the training data and therefore classifies the unknown test data worse, even though it was trained longer or more extensively. Underfitting describes the opposite effect; a model that is not yet optimized well enough with the training data (Goodfellow et al., 2016).

2.2.2. Unknown dataset procedures

In addition to the mathematical metrics for calculating the respective result values, the input data is crucial for evaluating the models. The introduced metrics can be determined with any data set, only the relevance varies with the selected datasets as these can be known or unknown. From literature three different ways of analyzing the trained model with unknown data are identified and illustrated in Figure 2. These types are an excerpt of possible procedures and are presented below.

The standard method splits the entire dataset randomly into training and test sets and is called **hold-out** (Bruce et al., 2021). It is based on a defined percentage, which usually varies between 10 % and 20 %. As the name suggests, the training data is used to train the model, whereas the test set is only applied for evaluation at the end. So, no further optimization is possible after the test data is submitted, otherwise, the data is considered as training. An enhancement of this idea is k-fold cross-validation (Bruce et al., 2021), in which the model is trained and tested with different fractions as test datasets. The entire database is divided into k subsets and then one of the k subsets is used for testing the model, while the remaining (k-1) subsets are applied for training. This process is repeated k times until each partial dataset has served once as an unknown test sample. The mean value of all evaluations is calculated, thus ensuring that a particularly simple test set did not lead to good results by accident. Another solution is the exact selection of training and test datasets, which often occurs for benchmark datasets. This is intended to minimize the random factor that is inevitably integrated into the evaluation when datasets are selected arbitrarily.

Besides the traditional hold-out evaluation, the model **robustness** is also crucial for its application success. The evaluation of robustness involves not only simple unknown data but actively manipulated data to artificially complicate the recognition. The goal is to develop a model that can deal with uncertainties or even modified data.

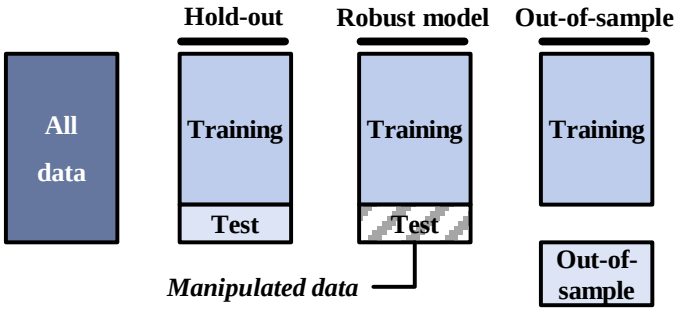

Figure 2. Different procedures that deal with unknown data in evaluation of Machine Learning models according to (Bruce et al., 2021; Malik et al., 2020; Dubrov, 2023)

This ability is particularly relevant for applications in real-life situations, such as financial transactions or autonomous driving cars. The underlying idea is older than the current robust Machine learning trend and is based on robust statistics (Tukey, 1975), which aims to provide statistical conclusions despite corrupted data. Robust models should handle manipulations in the datasets. According to (Dubrov, 2023) these modifications include noisy data, distribution shifts and adversarial attacks. The first two types can occur primarily in the real application of models. Noisy data can contain deviations, errors, missing values or many unimportant data points. Distribution shifts are primarily defined as situations where the training dataset does not reflect the distribution from the application case. Different conditions apply to adversarial attacks, in which the evaluation data is actively manipulated to deceive the model and thus expose its weaknesses. The types of perturbation are diverse, from simple lp deviations (Szegedy et al., 2013), translation and rotation of images (Engstrom et al., 2018) to placing stickers on real traffic signs (Eykholt et al., 2018). Special toolboxes to check for these types of attacks have already been developed by (Rauber et al., 2017; Engstrom et al., 2019).

One starting point for improving robustness is the complexity of the model. Less complex models can be more robust against perturbation, but tend to be less accurate overall, according to (Abu-Mostafa et al., 2012). A study with Convolutional Neural Networks (CNN) on image recognition conducted by (Su et al., 2018) reveals similar results. The comparison tested 18 different CNN architectures and demonstrated that less complex models are more robust to changes, but do not reach the accuracies of more complex models.

Another characteristic of Machine Learning models is their generalizability or generalization error. This metric provides feedback on the ability to adapt to new, unknown data and is consequently referred to as **out-of-sample error**, see Figure 2. According to (Malik, 2020) the out-of-sample test is the only relevant metric that really determines the success of trained models. However, the definition and determination of outliers is highly dependent on the domain, application and data availability. For example, out-of-sample performance is frequently employed in time series analyzes by evaluating the model with new and unknown time periods (Chu et al., 2023). A continuation of the generalization idea is testing the models with out-of-distribution (OOD) datasets. According to (Yang et al., 2021; Yang et al., 2023), these datasets should represent both the covariate shift and the semantic shift and thus enable a better conclusion about the generalizability of the model. The covariate shift rather describes the changed distribution within the classes, whereas the semantic shift creates differences between classes, all compared to the original training dataset. (Yang et al., 2023) define different categories for the datasets depending on the degree of shift expression, with all falling under the out-of-sample description. The different classes are training, covariate shifted, near-OOD, and far-OOD, which are all illustrated in Figure 3.

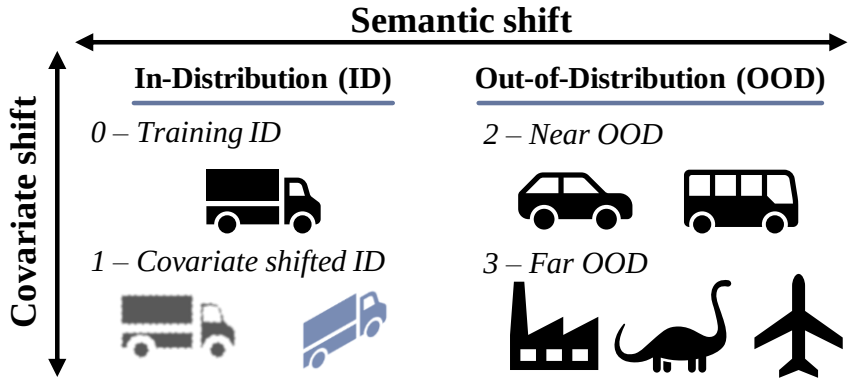

Figure 3. Example of different out-of-sample dataset categories depending on the shift type according to (Yang et al., 2023)

3. Research gap and approach

3.1. Research gap and question

In contrast to the academic environment, when applying trained models in the industrial setting, the input data cannot be accurately anticipated. Therefore, the consideration of the out-of-sample capability is extremely important. The general approaches were presented in the previous chapter, but the specific transfer to the engineering environment is missing. More and more data-driven methods are developed in the product development domain for decreasing the time to market or improving the user acceptance (Banerjee, 2021 and Ragani et al., 2023), making it increasingly necessary to test transferability on unknown datasets. This publication intends to address this gap for the plausibility detection method for FE simulations through answering the following research question: To what extent can trained Deep Learning models for detecting plausibility in FE simulations be transferred to unknown simulations and what procedure parameters influence the detection results?

3.2. Testing procedure

The basic procedure for answering the question is displayed in Figure 4. The process is based on the typical testing of trained models and is divided into the areas of data-generation, -preparation, model selection and evaluation. The relevant influential parameters for the individual steps are highlighted in the blue dashed boxes in the Figure. The initial step for the data generation is to develop an adaption of the established out-of-sample categories to the FE simulations domain, which requires determining the impact features on FE simulations and then specifying the gradations of these criteria. The data for the respective categories is then generated via parameter studies, whose structure and resulting data creation are examined in section 4. The dataset then serves as the basis for answering the research question. Different established CNN architectures as well as normalization strategies for the transformed matrices are also tested to determine their influence on the recognition accuracy. The evaluation and discussion of the obtained result are presented in sections 5 and 6. Concluding the paper, the possibilities for adapting the overall procedure to be better suited for unknown simulations are investigated.

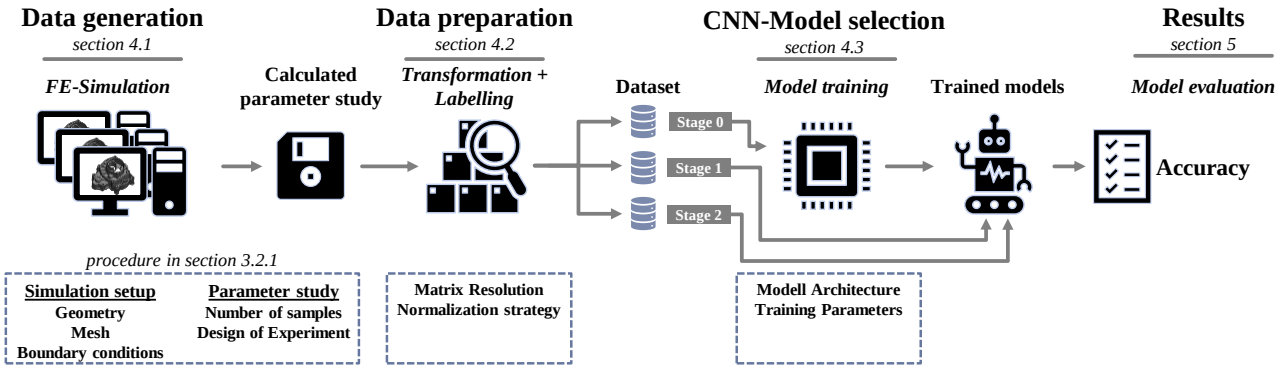

Figure 4. Overview of the procedure and the parameters for testing the generalizability of the plausibility check

3.3. Adaption of out-of-sample testing for FE simulation

The principle of in- and out-of-distribution data must be mapped to the domain of the FE simulations so that trained plausibility detection models are examined for their generalizability. The testing approach is orientated on the basic preprocessing steps in an FE analysis, which involves selecting and importing the geometry, defining supports and boundary conditions, meshing the component and assigning material parameters. Table 1 provides the list of the factors and their individual characteristics, which are classified into the categories of (Yang et al., 2023), whereby the degree of similarity stage 0 corresponds to the existing training data. The first degree of deviation corresponds to the covariate shift and contains new, unknown parameters for the existing simulations in the dataset. The possible parameters for the geometry, boundary conditions and meshing are newly selected and are therefore not available in the training dataset. In stage 2, similar or related geometries are applied instead of the original components. For this purpose, derivatives are created for each of the parts from the original dataset. New types of material are also used e.g., plastics instead of metals. The boundary conditions rely on the geometry and operation conditions and are therefore not included in the training database. In addition to changing element sizes, adaptive meshing is activated to create non-uniform meshes. This level corresponds to the near out-of-distribution class, as completely different but related parts are employed compared to the original dataset. The last stage describes simulations that are very far away from the original dataset; in the context of structural mechanics simulations. This means new components of a different Opitz code (Opitz, 1970) and also completely new load types. Other forms of simulation such as module analysis or Computational Fluid Dynamics could be categorized at this level. The large number of possible combinations is a problem when implementing the testing procedure, as new components are required for the geometric changes, making it impractical to test all full factorial combinations. However

several different parts must be tested, as otherwise, one geometry could work well or badly by chance. Therefore, the different stages are tested as a whole and no combinations (e.g., geometry stage 1; mesh stage 0; material stage 2 etc.) between the stages are considered.

Table 1. Adaptation of the out-of-sample testing for plausibility detection in FE simulation

Adapted generalization stages for FE simulations				
Stage	**0 – Training ID**	**1 – Covariate shift ID**	**2 – Near OOD**	**3 – Far OOD**
Geometry	Same part parameters	Altered part parameters	Similar part	New Opitz-class
Boundary condition	Same load case	Altered load case parameters	Altered point of load application	Unknown load case
Mesh	Same mesh	Altered element size	Adaptive meshing	
Material	Same material	Same material class	Altered material class	

4. Application study

4.1. Dataset

The transfer of the described generalization testing procedure to the existing method of plausibility detection with its dataset is illustrated in the overview in Figure 5. The image demonstrates the respective parts for the parameter studies, arranged according to the abstraction level. The intention is to test each stage separately with multiple components so that a conclusion about the generalizabilty can be made. In stage 1, the same components are used as in the training dataset, but with new parameters for the mesh, geometry and loads. This method was realized for four out of five components from the training set, except for the crankshaft because there was a compatibility problem between the CAD and FEM software. The next deviation category (stage 2) requires new components that are close to the training data but still vary noticeably. The similarity can be represented via the part function, for example, the bicycle rim (derived from car rim) and the brake pedal (derived from brake lever), or via the geometric similarity as in the case of the drone frame or the shaft. Possible components were also defined for stage 3, but these differed significantly from the original geometries. The general idea here was to remain in the product category but to select new components from it. For example, in the field of machine elements, a gearwheel is suitable for a drive shaft, in the suspension of a car a wishbone is close to a wheel rim and in the frame of a bicycle a spring is next to a rocker arm. The additional modification parameters such as material, mesh and boundary conditions vary also for each parameter study. The structure of these simulations is explained in the following section with a brief explanation of the basic structure of the simulation. All are structural-mechanical simulations of components that are turned into parameter studies with different experimental designs. All parameter studies of the test datasets were carried out with Ansys Workbench, which was connected to PTC Creo. The training dataset of stage 0 is taken from (Bickel et al., 2023) and is only used to train the models in this study. For the stage 1, existing simulations were utilized with new parameters according to the presented approach. In contrast, all components for stage 2 were newly designed and the associated simulations generated.

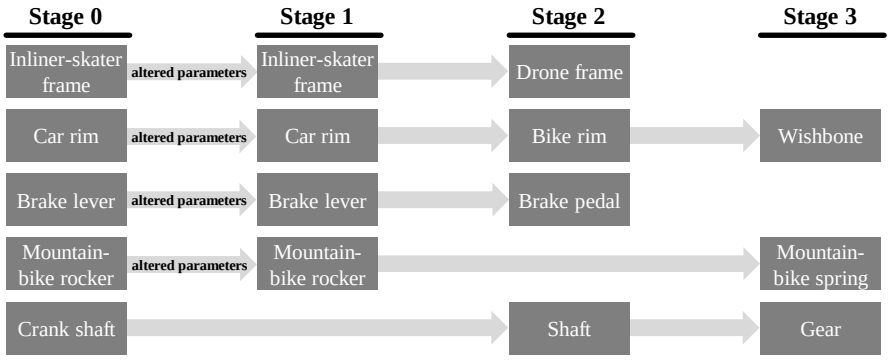

Figure 5. Overview of the simulation studies set up, including the respective specification level and components

4.1.1. Stage 1 – Covariate in-distribution

The simulation studies of stage 1 are based on the preliminary work of (Bickel et al., 2023) and comprise four simulations. The exact setup and the associated load cases are explained in the publications. In total, over four thousand new simulations were calculated, with almost equal distribution between plausible and non-plausible simulations. All results combined have a memory consumption of approx. 270 GB, with an exact breakdown for each component listed in Table 2. The cause

of a non-plausible simulation can be the geometry, the mesh or the load case. In all studies, parameters for these three possible causes are defined via a d-optimal design of experiment (DOE) plan. As described in section 3.2, parameter values that are not included in the previous training data were chosen.

Table 2. Overview of the numbers of simulations in the stage 1 dataset

Part name	Number of simulations	Plausible	Non-plausible	Storage space
Mountain bike rocker	1,026	540	486	17.0 GB
Inliner frame	1,008	432	576	17.5 GB
Brake lever	1,045	563	482	88.7 GB
Car rim	952	400	552	144 GB
Whole dataset	**4,031**	**1,935**	**2,096**	**267.2 GB**

4.1.2. Stage 2 – Near out-of-distribution

A total of four new CAD geometries were created for the analysis of stage 2. The models are similar to the parts from the training dataset either geometrically or functionally and include a drone frame, a brake pedal, a shaft and a bicycle rim. The setup of the individual simulations and the respective component geometry are shown in Figure 6. The load cases for all components represent a realistic product development step, so that all simulations have reference models for the boundary conditions.

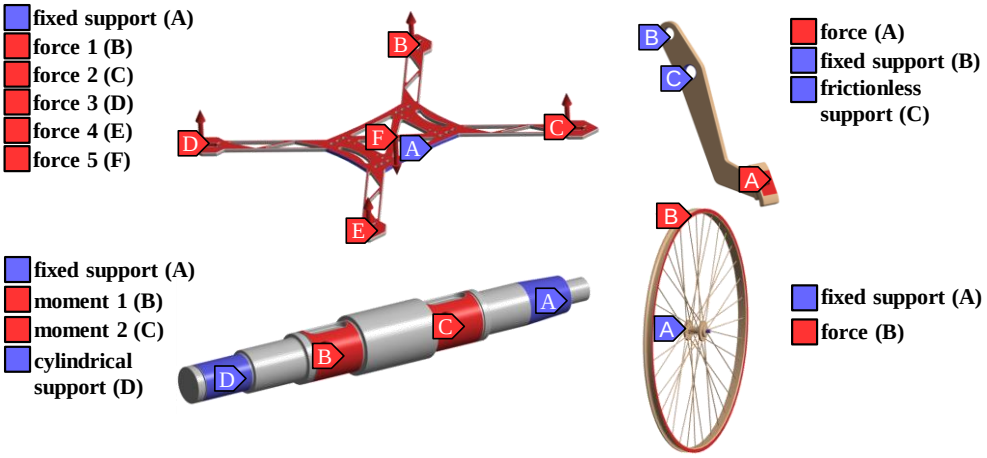

Figure 6. Simulation setup for the new components in the stage 2

The shaft was designed for a pedestrian stacker which is used in industrial environments for transportation and lifting of heavy items.. The constraints were defined via a requirement specification (lifting weight 1,000 kg) for the pedestrian stacker and the shaft was fitted with bearings and loads according to the task. The motor is connected to the shaft via a tapered press connection and the sprockets transmit the torque via feather keys. The other end of the shaft is supported by a floating bearing.

The next component is a drone frame, which serves as the central structure that carries all the important components such as the battery, control system, motors and cameras. Previous studies for 3D-printed drones (Shelare et al., 2021) and quadcopters (Ahmad et al., 2021) provided the foundation for the boundary conditions, which simulate the situation of a drone taking off. The operating mode is composed of the thrust of the motors, which is realized via a force on each of the four arms of the frame, and the weight of the electronics, batteries and camera.

The brake pedal is an essential part of the vehicle control and braking system and serves as the third model. The simulation setup and the associated values were again derived from the existing work of (Kharde et al., 2021; Sargini et al., 2020). The force is applied to the tread surface and transferred to the brake cylinder via the upper bore, which corresponds to the fixed support in the FE simulation. The pedal is supported by a frictionless bearing on the second hole.

The last component is a bicycle rim. It is the link between the bicycle frame and the tire and must withstand high loads during operation, as all unevennesses and ground conditions are transferred directly to it. Standards were developed for the actual testing of bicycle rims, which are summarized in (ISO 4210-7:2023). The norm defines a test setup that fixes the rim to a clamping device and then applies a static force to a spoke perpendicular to the plane of the wheel. This setup was incorporated into the FE system and reproduced with the corresponding values from the standard.

From the individual simulation setups numerous datapoints are created through paramter studies. All possible parameters (mesh, load, material, and geometry) were again integrated into a d-optimal DOE. Geometric parameters are assigned to all parts and allow the CAD model to automatically adapt. An example of the different parameters used on the brake pedal is the thickness and length of the pedal, as well as the width and height of the tread surface. The aim was always to calculate 1,000 design points per study. Afterwards, an Ansys Parametric Design Language (APDL) script saved the results for each simulation as a text file, which includes the stresses, deformations, boundary conditions and mesh. After executing all simulations, a dataset of approx. 3,000 simulation results was created for stage 2, of which 1,358 are plausible and 1,653 are classified as non-plausible, as stated in Table 3. The memory requirement is almost 1 TB, whereby the required memory space depends heavily on the simulation setup, geometry and meshing. For all dataset, the labelling was carried out using a combination of automated (e.g. rule based for too high force) and manual marking.

Table 3. Overview of the numbers of simulations in the stage 2 dataset

Part name	Number of simulations	Plausible	Non-plausible	Storage space
Shaft	1,008	460	548	287 GB
Drone frame	1,033	416	617	135 GB
Brake pedal	646	352	294	81.8 GB
Bike rim	324	130	194	416 GB
Whole dataset	**3,011**	**1,358**	**1,653**	**919.8 GB**

4.2. Data preparation

Before the generated data is processed by a Deep Learning model, it must be transformed into matrices. For this purpose, certain parameters are specified for the training and subsequent test dataset so both can be evaluated by the model. The first parameter is the resolution of the matrices, which was set to 100 × 100 as it is the best compromise between accuracy and computational load. Furthermore, the specific pre- and postprocessing matrices from the results must be chosen. The selection of matrices was based on the idea of a wide application range for plausibility detection. As a result, 26 different matrices are generated for one simulation: nodes, fixed translation and rotation in the X-, Y- and Z- directions; force, external force and pressure in the X-, Y- and Z- directions; moment about the X-, Y- and Z- axes; positive and negative displacements in the X-, Y- and Z- directions and the equivalent to von Mises stress. Once the calculation results are successfully transformed into the matrices, they are subsequently normalized. Two options are tested in this paper: First, the specific normalization from (Spruegel et al., 2021) and second no normalization of the matrices. The intention is to investigate whether a specific adaptation of the normalization to FE simulations has a positive effect on detection.

4.3. CNN-Architecture

As described in the state-of-the-art section, a CNN is required for the classification of plausibility. Several popular CNN architectures are utilized in this study and their ability to recognize unknown simulations is tested. The general problem for the application of CNNs as part of the plausibility check is the modified model input. Normally, CNNs have an input of X × X × 3, as they were developed for RGB images. In the case of the plausibility check however, this input changes to X × X × 26, which means that the structure and the respective layers must be adapted. In preliminary work in this area by (Bickel et al., 2022; Bickel et al., 2023) several established and readily available CNN architectures were modified for this new application domain, although the employed CNNs differ in terms of structure and complexity. The principal architecture of the Deep Learning models was inspired by: Inception-V3 (Szegedy et al., 2015), VGG16 (Simonyan and Zisserman, 2014), DenseNet (Huang et al., 2017), MobileNet (Howard et al., 2017) and ResNet (He et al., 2016). The exact adaptations and changes to the structure of the various CNN architectures are described in detail in (Bickel et al., 2023). For the training, the data was automatically partitioned into training, validation and test datasets according to the 72 % - 8 % - 20 % principle. All models were trained on the identical computer server, which is equipped with two AMD EPYC 7643 processors, 256 GB RAM and two Nvidia A40 (46 GB) graphic cards.

5. Results

After all models were successfully trained with the same training dataset (stage 0), their accuracy is determined. The overall results for the different stages are displayed in the bar chart in Figure 7, whereby a distinction is made between the normalization strategies and the employed networks.

The evaluation of level 0 demonstrates that normalizing the data leads to better accuracy for all networks, besides the ResNet. The influence of the network architecture is thus reduced by the specific preparation of the data. Overall, the models with the own normalization achieve a very high level of accuracy with the test dataset, with an average value of 0.985 across all models. In contrast, the accuracy is considerably lower if the data is not normalized, with a mean value of only 0.953. These results also reflect the findings from previous publications (Bickel et al., 2023 and Spruegel et al., 2021), where it was stated that a specific normalization leads to an improved recognition of plausibility.

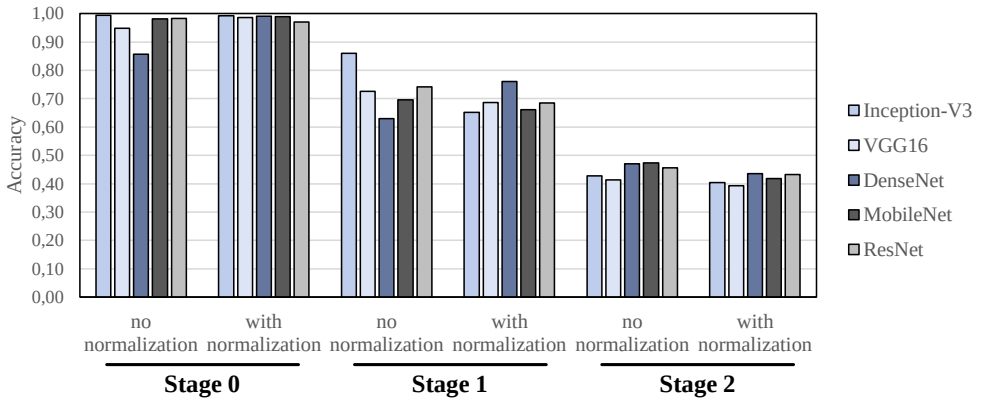

Figure 7. Mean classification accuracy over all parts split by stage, normalization strategies and architecture

Evaluating stage 1 of modification reveals a different picture, as the non-normalized matrices achieved significantly higher values. In contrast to level 0, the CNN architecture has a greater influence on accuracy, with the two complex architectures Inception V3 and ResNet as the best representatives. Only the DenseNet improved using the normalized dataset, reflecting the stage 0 results where the model improved most by changing the normalization strategy. Overall, the Inception-V3 was able to obtain a very high accuracy of 0.860 for the unknown dataset, with the values for some parts even exceeding 0.940, as the evaluation in Figure 8 illustrates. Looking at the different parts, it is particularly noticeable that the car rim was difficult to classify for all models. The reason could be that in addition to the load and element size, the rim and also the brake lever vary geometrically from the training dataset and are therefore more difficult to classify. This phenomenon is also evident in the ResNet model, which achieves significantly better values for the mountain bike rocker and the inline frame than for the other two components. The VGG16 and MobileNet are at a similar level and tend to reach around 0.70 accuracy for each individual part.

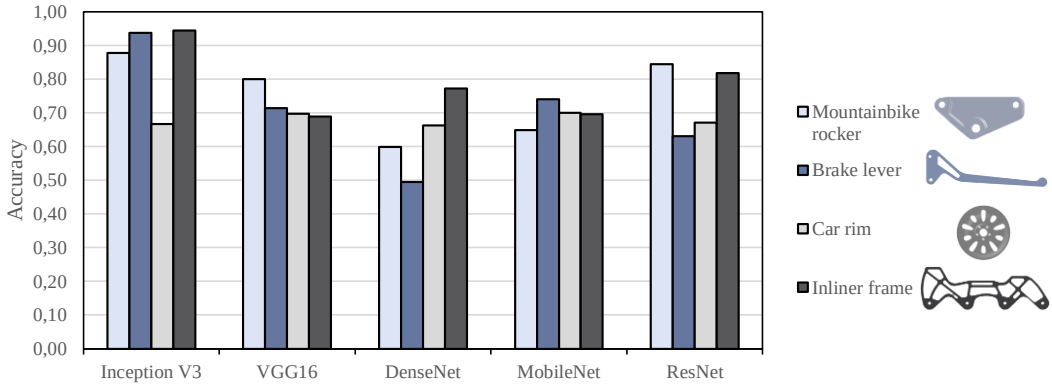

Figure 8. Classification accuracy for stage 1 with no normalization, split by part and architecture

Stage 2, on the other hand, shows a significantly inferior performance for plausibility detection as apparent on the right side of Figure 7. All tested models achieve very low accuracy values below 50 %, which would be achieved by randomly guessing. Consequently, further examination of stage 3 results makes little sense. The causes and reasons for the obtained values are discussed in the next section and the necessary action steps for the plausibility detection method are derived from this.

6. Discussion

In general, the recognition accuracy decreases as the test data moves further and further away from the training data. The results obtained for stages 0 and 1 are promising and indicate that the method is robust against this kind of deviations from the training data. That seems to be the current limit of the applied method as no reasonable accuracy could be achieved for stage 2, which answers the stated research question. The study showed that the model selection has a major influence on the accuracy of data outside the parameter space of the training data. Contrary to the common principle that complex models lead to a higher out-of-sample error (Abu-Mostafa et al., 2012), the results reveal that the opposite effect occurred. In comparison, the parts themselves have a greater effect on the recognition accuracy, which may be due to the limited availability of geometrically parameterized parts in the training dataset and the characteristics of the projection method.

Another cause for the poor results is the definition of plausibility, as this was previously described as "likely valid" simulation results. This creates a range of uncertainty between the plausible and non-plausible labels, where the assignment of plausibility is no longer conclusive. One possible approach would be to include this sector as a class and categorize it as "no reliable statement possible". In addition, stage 2 components were possibly too far from the original data and should therefore be assigned to stage 3 rather than stage 2. In the future, an intermediate stage should be introduced, which consists of modifications or adaptations of stage 2 parts, for example, a crankshaft for a four-cylinder engine.

In addition to the plausibility definition and the part-choice in the unknown datasets, the training of the models must likewise be discussed. One potential idea could be to reduce the number of input matrices to make the model leaner and therefore more robust. Currently 26 input matrices are used, which are divided into pre- and postprocessing results. Using only the eight postprocessing matrices could be sufficient for assessing plausibility. In addition, the training parameters offer a good opportunity for more generalizable models. For example, the training duration could be shortened by reducing the validation patience, thus lowering the probability of overfitting during model training. Furthermore, additional drop-out layers can be integrated into the models, which automatically delete a percentage of learned weights and should lead to more general models.

Also noticeable in stage 2 was the tendency of the trained models to generally determine "non-plausible" as the result of the classification when given unknown simulations. For a hypothetical industrial application as described in (Bickel et al., 2019), this circumstance is quite beneficial, as the model warns the user and does not define unknown or even faulty simulations as plausible. In this scenario, an experienced calculation engineer would have to check the simulations in the event of a non-plausible result and thus verify the calculated results. This engineer could then include the simulation results with their corresponding labels in the database and reduce the problem for subsequent simulations of similar components.

7. Conclusion and outlook

In summary, this publication presents and applies a procedure for testing the generalizability of the plausibility check method to unknown simulations. First, different possibilities for testing trained models were analyzed. Subsequently, the findings were transferred to the domain of FE simulations by defining different stages of abstraction for comparison. A total of eight simulation studies with a combined volume of 7,042 calculated results were generated to allow a reliable conclusion on the current generalizability of the method. This unknown dataset was tested on a total of five different CNN architectures, all adapted to the task of using simulation matrices as model input. The evaluation reveals that the current plausibility detection method for FE simulations recognizes the stage 1 well, but achieves poor accuracy for the stage 2 dataset, with the possible causes and reasons discussed and analyzed in detail. The aim of future work is to improve the procedure so that greater accuracy is achieved for unknown simulations. The opportunities are manifold, starting with the training data and associated DOE, the training and architecture of the models, the preparation and selection of the simulation outcome for the projection, the definition of plausibility and finally the structure of the unknown datasets. Furthermore, the underlying method could be revised so that a high level of accuracy can ultimately be achieved even with unknown components from the third deviation stage.

Acknowledgement

The authors thank the German Research Foundation for funding this research under grant number WA 2913/47-1.
The authors thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Titan GPU.

References

Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.-T., 2012. Learning from data, 4th ed. AMLBook New York.

Ahmad, F., Kumar, P., Patil, P.P., Kumar, V., 2021. FEA based frequency analysis of unmanned aerial vehicle (UAV). Materials Today: Proceedings 46, 10396–10403. https://doi.org/10.1016/j.matpr.2020.12.740.

Banerjee P., 2021. How AI and ML are Changing Simulation, ANSYS Blog. https://www.ansys.com/blog/how-ai-and-ml-are-changing-simulation (accessed 18.04.2024).

Bennett, J., Creary, L., Englemore, R., Melosh, R., 1978. SACON: A knowledge-based consultant for structural analysis. Stanford Univ Calif Department of Computer Science.

Bickel, S., Goetz, S., Wartzack, S., 2023. Detection of Plausibility and Error Reasons in Finite Element Simulations with Deep Learning Networks. Algorithms 16, 209. https://doi.org/10.3390/a16040209.

Bickel, S., Schleich, B., Wartzack, S., 2022. Resnet networks for plausibility detection in finite element simulations, in: DS 118: Proceedings of NordDesign 2022, Copenhagen, Denmark, 16th - 18th August 2022. NordDesign 2022, pp. 1–10.

Bickel, S., Spruegel, T., Schleich, B., Wartzack, S., 2019. How Do Digital Engineering and Included AI Based Assistance Tools Change the Product Development Process and the Involved Engineers. Proc. Int. Conf. Eng. Des. 1, 2567–2576. https://doi.org/10.1017/dsi.2019.263.

Bruce, P.C., Bruce, A., Gedeck, P., 2021. Praktische Statistik für Data Scientists: 50+ essenzielle Konzepte mit R und Python, 1st ed. O'Reilly, Heidelberg, 356 pp.

CEN European Committee for Standardization, 2023-01-00. Cycles - Safety requirements for bicycles - Part 7: Wheel and rim test methods (ISO 4210-7:2023).

Chu, B., Qureshi, S., 2022. Comparing Out-of-Sample Performance of Machine Learning Methods to Forecast U.S. GDP Growth. Comput Econ 62, 1–43. https://doi.org/10.1007/s10614-022-10312-z.

Dubrov V., 2023. Understanding Machine Learning Robustness: Why It Matters and How It Affects Your Models. https://medium.com/@slavadubrov/understanding-machine-learning-robustness-why-it-matters-and-how-it-affects-your-models-5e2cb5838dab (accessed 17 January 2024).

Dumbach, P., Schwinn, L., Löhr, T., Elsberger, T., Eskofier, B.M., 2023. Artificial intelligence trend analysis in German business and politics: a web mining approach. International Journal of Data Science and Analytics. https://doi.org/10.1007/s41060-023-00483-9.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., Tsipras, D. Robustness (Python Library). https://github.com/MadryLab/robustness (accessed 17.01.2024).

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A., 2018. A rotation and a translation suffice: Fooling cnns with simple transformations.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D., 2018. Robust Physical-World Attacks on Deep Learning Visual Classification, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.

Goodfellow, I., Courville, A., Bengio, Y., 2016. Deep learning. The MIT Press, Cambridge, Massachusetts, 775 pp.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2017. Densely Connected Convolutional Networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Ippen, H., Bach, M., 2019. Verkehrsschild-Erkennung: Test Ungenaue Schilder-Erkennung. https://www.autozeitung.de/verkehrsschild-erkennung-test-195733.html (accessed 17 January 2024).

Kharde, A., Khande, R., Khedekar, S., Manchekar, J., Anavkar Jayesh S., 2021. Design and mass optimization of brake pedal using topology optimization technique. International Research Journal of Engineering and Technology (IRJET).

Malik, M.M., 2020. A Hierarchy of Limitations in Machine Learning, 68 pp.

Opitz, H., 1970. A classification system to describe workpieces. Pergamon Press, Oxford, England, 213 pp.

Powers, D.M.W., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, 27 pp.

Ragani, F.-A., Stein P., Keene R., Symington I. 2023. Unveiling the next frontier of engineering simulation, McKinsey and NAFEMS, https://www.mckinsey.com/capabilities/operations/our-insights/unveiling-the-next-frontier-of-engineering-simulation#/ (accessed 18.04.2024).

Rauber, J., Brendel, W., Bethge, M., 2018. Foolbox: A Python toolbox to benchmark the robustness of machine learning models.

Sargini, M.I., Masood, S.H., Palanisamy, S., Jayamani, E., Kapoor, A., 2020. Finite element analysis of automotive arm brake pedal for rapid manufacturing. IOP Conf. Ser.: Mater. Sci. Eng. 715, 12020. https://doi.org/10.1088/1757-899X/715/1/012020.

Shelare, S.D., Aglawe, K.R., Khope, P.B., 2021. Computer aided modeling and finite element analysis of 3-D printed drone. Materials Today: Proceedings 47, 3375–3379. https://doi.org/10.1016/j.matpr.2021.07.162.

Shivaditya M. V., Bugiotti F., and Magoules F., 2022. Point-Cloud-based Deep Learning Models for Finite Element Analysis, arXiv.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://doi.org/10.48550/arXiv.1409.1556.

Spruegel, T., Bickel, S., Schleich, B., Wartzack, S., 2021. Approach and application to transfer heterogeneous simulation data from finite element analysis to neural networks. Journal of Computational Design and Engineering 8, 298–315. https://doi.org/10.1093/jcde/qwaa079.

Spruegel, T., Hallmann, M., Wartzack, S., 2015. A concept for FE plausibility checks in structural mechanics, in: Proceedings of the NAFEMS World Congress, San Diego, CA, USA.

Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.-Y., Gao, Y., 2018. Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer vision - ECCV 2018: 15th European conference, Munich, Germany, September 8-14, 2018 : proceedings. Springer, Cham, pp. 644–661.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2014. Intriguing properties of neural networks. arXiv.

Tukey, J.W., 1975. Mathematics and the Picturing of Data. Proceedings of the International Congress of Mathematicians, Vancouver, 1975 2, 523–531.

Woyand, H.-B., Goldhammer, L., Roj, R., 2012. A knowledge based assistance system for finite element analysis, in: 2012 15th International Conference on Interactive Collaborative Learning (ICL). IEEE.

Yang, J., Zhou, K., Liu, Z., 2023. Full-Spectrum Out-of-Distribution Detection. Int J Comput Vis 131, 2607–2622. https://doi.org/10.1007/s11263-023-01811-z.

Yang, J., Zhou, K., Li, Y., Liu, Z., 2022. Generalized Out-of-Distribution Detection: A Survey.

Zhou, Z.-H., 2021. Machine learning. Springer Singapore; Springer, Singapore, 458 pp.

Contact: Sebastian Bickel. Friedrich-Alexander-Universität Erlangen-Nürnberg, Engineering Design, Martensstrasse 9, 91058 Erlangen, Germany, bickel@mfk.fau.de