Glass Struct. Eng.
https://doi.org/10.1007/s40940-022-00167-z

PROJECTS & CASE STUDIES PAPER

SoundLab AI-Machine learning for sound insulation value predictions of various glass assemblies

Michael Drass · Michael Anton Kraus · Henrik Riedel · Ingo Stelzer

Received: 7 December 2020 / Accepted: 21 December 2021

Abstract Modern architecture promotes a high demand for transparent building envelopes and especially glass facades. Commonly, facades are designed to fulfill a multitude of objectives such as superior aesthetic appearance, a higher degree of weathering reliability, quick installation, high transparency as well as economic and ecologic efficiency. For such glazing applications, often an assessment of acoustic properties and especially sound insulation abilities are required. Because of the complexity of such an experimental or computational investigation given the framing systems and glass unit compositions, a reliable and fairly accurate estimation of sound insulation properties of such systems becomes time-consuming and demanding. This paper provides a Machine Learning (ML) based estimation tool of acoustic properties (weighted sound insulation value R_W, STC and $OITC$) of different glazing set-ups. A sufficiently rich database was used to train several machine learning algorithms. The acoustic properties are determined by comparing the third-octave or octave band spectrum of the sound reduction index with a reference curve (typical curve for solid construction elements) specified in the standard DIN EN ISO 717-1. Sound insulation values can currently only be determined by complex and expensive experimental investigations or numerical simulations for certain glass set-ups. Hence, there is no efficient tool for convenient and reliable estimation of the sound insulation performance of glazing systems available at the moment. To this end, the engineering team led by the authors conducted extensive studies on various glazings consisting of different glass assemblies with varying glass, cavity and interlayer thicknesses and different types of interlayer and gas fillings. Based on our research outcomes, a comprehensive web-based prediction program, the so-called AI Tool, has been developed recently. This program can provide a quick analysis and accurate prediction of arbitrary glazing set-ups, interlayers and glazing infills. A series of laboratory tests were conducted to validate the predictions by the AI Tool. The goal of this program is to provide designers, engineers, and architects an effective and economically efficient tool to facilitate the design w.r.t. acoustical properties.

Keywords Artificial intelligence · Machine learning · Glass · Acoustic properties · Sound insulation value

M. Drass (✉) · M. A. Kraus · H. Riedel
M&M Network-Ing UG (haftungsbeschränkt), Lennebergstraße 40, 55124 Mainz, Germany
e-mail: drass@mm-network-ing.com
https://www.mm-network-ing.de

M. A. Kraus
e-mail: kraus@mm-network-ing.com
https://www.mm-network-ing.de

I. Stelzer
Kuraray Europe GmbH, Philipp-Reis-Straße 4, 65795 Hattersheim am Main, Germany
e-mail: Ingo.Stelzer@kuraray.com

1 Introduction

In building acoustics, so-called single-number metrics for the sound insulation value have become established for the evaluation and comparison of facade products in terms of glazing systems and different glass assemblies. A glass assembly in this context describes the composition of the glazing without, however, taking into account the frame or the substructure. They allow the characterization of the sound insulation of a building component (e.g. laminated glass (LG), insulating glass units (IGU) etc.) without considering the frequency dependency, which significantly increases comprehensibility for non-experts and simplifies the formulation of technical requirements for glasses and glazing systems. The weighted sound insulation value R_W used in Europe as well as the sound transmission class (STC) and the out-/indoor transmission class $OITC$ values used in the USA are determined from sound transmission loss (STL) measurements at certified laboratories by comparing the third-octave in the range from 100 to 3 150 Hz) or octave (in the range from 125 to 2 000 Hz) band spectrum of the sound reduction index with a reference curve (typical curve for solid construction elements) specified in the standard DIN EN ISO 717-1.

To this day, the determination of the sound insulation values can currently only be determined by complex numerical analyses or expensive experimental tests. This raises the question of whether it is possible to use artificial intelligence (AI) and machine learning (ML) methods to develop a robust prediction tool to determine the sound insulation value for arbitrary glass assemblies. Therefore, this case study presents for the first time the so-called SOUNDLAB AI tool, which will be abbreaviated as AI tool or ML tool in the remainder of this paper.

Based on prior experience of M&M Network-Ing UG (haftungsbeschränkt) in that field from an academic and industry-related perspective (cf. (Kraus and Drass 2020a, b; Kraus and Taras 2020) etc.), the requirements in the project between Kuraray Europe GmbH and M&M Network-Ing UG (haftungsbeschränkt) were to develop an AI-based prediction tool based on the database provided by Kuraray Europe GmbH. In addition, a requirement for the accuracy of the AI tool was set, namely an accuracy bandwidth of ±1dB. This means that the predictions by the tool must reach the value from the database with an uncertainty of ±1dB.

In summary, this case study reports on how the ML project between Kuraray Europe GmbH and M&M Network-Ing was set up with regard to the development of an AI-based tool for predicting the sound insulation value of any glass assemblies or set-ups.

2 Basics on acoustic properties of glass

Domestic noise is a high stress factor, particularly when it is experienced in the intimacy of the own homestead or in an working environment. With efficient glazing systems adapted in design to the acoustic demands of a certain stakeholder, it is possible to maintain a satisfactory environment for different purposes. Especially in early design stages, a quick evaluation and prediction of acoustic capabilities of a given glass set-up might be of high interest and crucial for the building owner or the tenants. Based on this circumstance, this paper presents a novel AI-based tool for evaluation of acoustic glass system properties in order to allow a reliable and efficient assessment and optimization of acoustic performance.

2.1 Influential Parameters of the glass assemblies on acoustic performance

From previous studies it is known that essential parameters influencing acoustic properties of glazing systems are the glass thickness, the use of laminated safety glass/special acoustic interlayers and the use of insulating glass with cavities filled with gases such as argon and krypton. In the following, the influencing factors mentioned for controlling the acoustic performance of glass assemblies are briefly outlined to give the reader a sense of the relevant parameters.

2.1.1 Glass thickness effect

Each material's sound absorption depends on its mass, stiffness and damping properties. For a single glass pane, the only effective way to increase its performance is to increase its thickness, as stiffness and damping cannot be changed. The acoustic transmission loss for a single glass pane, measured over a range of frequencies, varies with the thickness of the glass.

Although thicker glass tends to provide greater sound insulation, it can actually transmit more sound at certain frequencies. Each glass thickness has a weak frequency value, i.e. a frequency for which this glass is less sound-absorbing than the others. This value is called the critical frequency. For example, a 4 mm thick glass is fairly transparent for high frequencies around 3500 Hz (poor attenuation measured in dB); 6 mm thick glass is poor for frequencies around 2000 Hz; and 10 mm thick glass performs poorly at 1300 Hz. The higher the mass, the less problematic the critical frequency seems to be: 25 mm thick glass has no weak point. The critical frequency problem can also arise in an insulating glass unit consisting of two panes of the same thickness. At this frequency, the two panes are said to vibrate (resonate) together reducing the overall acoustic performance of the glass.

2.1.2 Laminated vs. monolithic glass

Considering laminated and monolithic glass, the sound transmission is more attenuated by a laminated glass than by a monolithic glass of the same mass. As an example, a laminated glass of 2+2 mm reduces sound at high frequencies much more than a monolithic glass of 4 mm thickness. It is because the effect of the critical frequencies disappears due to the sound attenuation provided by polyvinyl butyral (PVB). The same applies to the 3+3 mm laminated against the monolithic 6 mm. At low frequencies (traffic noise), however, the effect of polymer interlayer is less pronounced, but still positive.

2.1.3 Cavity effect

A standard double glazing unit does not reduce the sound transmission substantially more than a monolithic glass. The air space between the glass panes only becomes decisive in the case of really wide cavities. For increasing sound absorption, the ideal cavity width is 200 mm. At widths below (or above) 200 mm, the effect is less noticeable (although a wide cavity always works better than a narrower one). Double glazing with a 10 mm air gap behaves almost like a 20 mm air gap.

2.2 State-of-the-art for predicting sound insulation values of glass structures

According to Chen et al. (2019), DeGanyar et al. (2019), to this day there is no efficient tool available for convenient and reliable estimation of the sound insulation performance of fenestration systems. Still, laboratory testing of windows, facades and curtain walls is the most accurate way of evaluating acoustic performance of different glazings. On the other hand, empirical, analytical or numerical methods for at least preliminary evaluation of glass set-up w.r.t. their acoustic properties are of significant pragmatic value especially for early design choices within a building project.

So far, analytical models for estimating acoustic properties of single and multi-layer glasses with / without cavities exist (Chen et al. 2019), which however diverge unsatisfactorily from test results, especially for bounded, sealed and laminated glass. It is worth noting that the estimate can be above or below measured values, hence these models cannot be used for“worst case” predictions.

In most cases, simple linear or non-linear regression models (Kurra 2012) are used to estimate the sound insulation value of various glass assemblies or glazings. Although these models are very efficient, understandable and robust, they lack the ability to generalize for arbitrary glazings, arbitrary laminates, glass thicknesses, etc. The possible combinations of different glass assemblies are very large, so that regression models only cover a small range of the diversity mentioned above. This is where our general model comes in, which can make a robust and fast prediction for any glass thickness, interlayer thickness and cavity thickness, for both asymmetrical and symmetrical insulating glass. It can therefore be said that the present ML model is a generalisation of the above-mentioned regression models.

On the other hand, numerical approaches like the Finite Element Method (FEM) and Finite Element Analysis(FEA) can model both air-borne, structure-borne and the linked fluid-structure interaction behavior of window frames and glazing (DeGanyar et al. 2019) through explicitly computing the cavities of the analyzed glazing. So far it is known that FEA are more accurate in the low-frequency range due to meshing and time-stepping conditions. However, meaningful vibro-acoustic simulations requires an advanced level of expertise to set-up and judge the FEM model and results. An alternative to FEM is the use of the Statistical Energy Analysis (SEA) technique, which is often used in the automotive and aircraft industry. Due to its advanced mathematical framework, SEA is not very practical for efficient gathering of performance information on the acoustic behavior of a given glass system. For further details on currently researched hybrid approaches in that direction, the reader is referred to (DeGanyar et al. 2019).

3 SoundLab AI tool

3.1 Basics on artificial intelligence and machine learning

In this section, we define the essential terms of Artificial Intelligence (AI) and Machine Learning (ML) to enable the reader to understand the basic approach and methods used to develop the ML Tool. Since this paper is presenting a case study, the individual steps of the development of the ML tool are described in a textbook-manner. The data used cannot be disclosed for reasons of confidentiality, but the tool is available online to anyone free of charge.

Artificial intelligence and Machine Learning is the science of getting computers to learn and make predictions, without being explicitly programmed. AI is dedicated to the theory and development of computational systems capable of performing tasks that normally require human intelligence, such as visual perception, speech recognition, decision making and translation between languages. Machine learning is a subclass of AI, which enables systems to learn from given data without the need of explicit programming for a specific problem. The aim of ML is to generate artificial knowledge from experience, which is in this context and also generally data. ML algorithms build a mathematical model $\mathcal{M}$ to infer between quantities of interest (features; targets) based on data to make predictions or decisions without being explicitly programmed (Frochte 2019; Rebala et al. 2019; Chowdhary 2020; Murphy 2012). However, a fundamental prerequisite is that the knowledge gained from the data can be generalized and used to solve new problems, to analyses previously unknown data or to make predictions about unmeasured data (prediction). In general, ML also has a strong link to optimization, as learning problems are typically formulated as minimizing a loss function on a number of training examples (Bishop 2006; Goodfellow et al. 2016; Murphy 2012). Furthermore ML is closely related to statistics in terms of methods but differ in their goal of drawing population inferences from a sample (statistics) vs. finding generalization predictive patterns (Bzdok et al. 2018).

Two different learning types can be distinguished for ML: supervised and unsupervised learning (Mitchell 1997; Bishop 2006; Goodfellow et al. 2016; Frochte 2019). Only supervised learning and specifically the formulation of a regression model are addressed here, as the present AI tool is based on this special form of algorithms (cf. Fig. 1).

In supervised ML projects, it is essential to have a dataset $D = (x_n, t_n)_{n=1}^{N}$ with N observations, where the dataset must have feature/influence variables x_n and target/response variables t_n. Both variables can be continuous or discrete. While supervised learning aims to develop a predictive model $\mathcal{M}$ based on both influence and response variables, unsupervised learning learns a model based only on the features (clustering; dimensional reduction). In supervised learning, a distinction is made between classification and regression problems. While in the first case the response variables t_n can only assume discrete values, the response variables t_n are continuous in regression problems.

Focusing on the present case study, namely the presentation of the AI Tool, the problem at hand was described as a regression problem in a supervised learning task. The goal of solving a regression problem is to predict the value of one or more continuous target variables $\mathbf{t}$ given the value of a vector $\mathbf{x}$ of input variables. By using regression models, it is furthermore possible to catch non-linear and more complex dependencies between the in- and outputs. For further information it is referred to (Kraus 2019; Bishop 2006; Goodfellow et al. 2016; Mitchell 1997; Lee et al. 2018; Murphy 2012).

3.2 ML project for soundLab AI tool

A generally valid scheme of steps involved in a successful ML project is presented in Fig. 2. In the following, all steps related to the development of the ML tool are briefly presented. Therefore, different algorithms are analyses and evaluated in order to obtain the best possible result for the prediction of the weighted sound insulation value. For reasons of confidentiality, the data used may not be made available or printed, but the structure of the algorithms used will be explained. Additionally, the AI tool is available online free of charge for everyone.

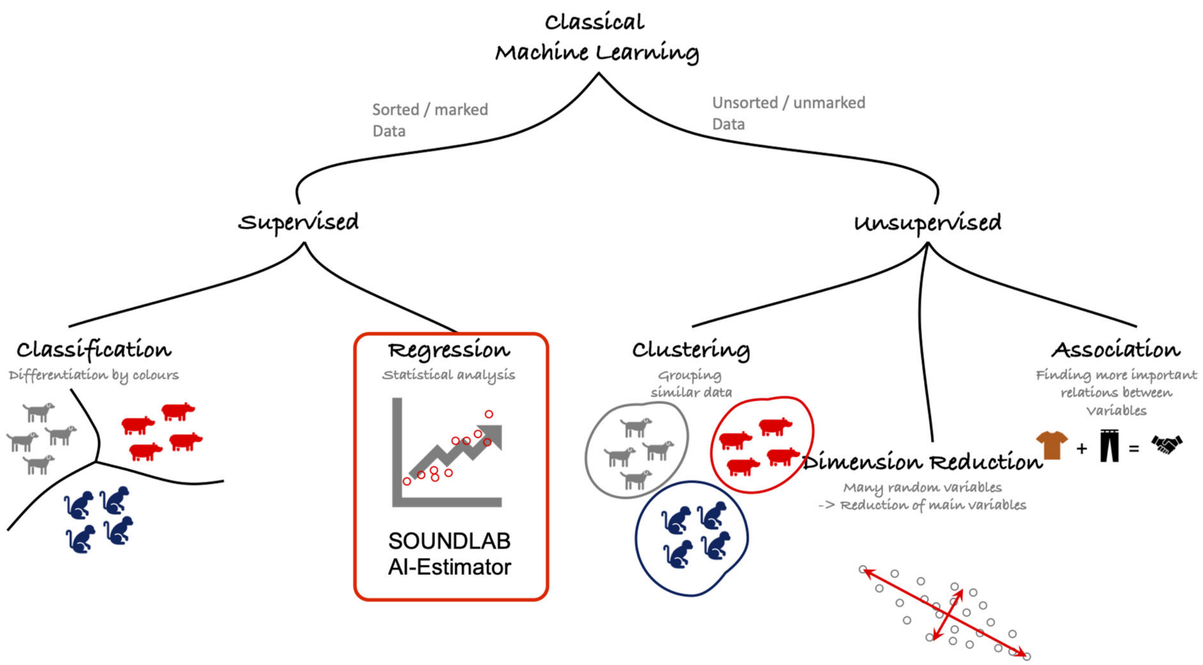

Fig. 1 Overview on the ML techniques with special focus on supervised regression models and the AI Tool, from (Kraus and Drass 2020a)

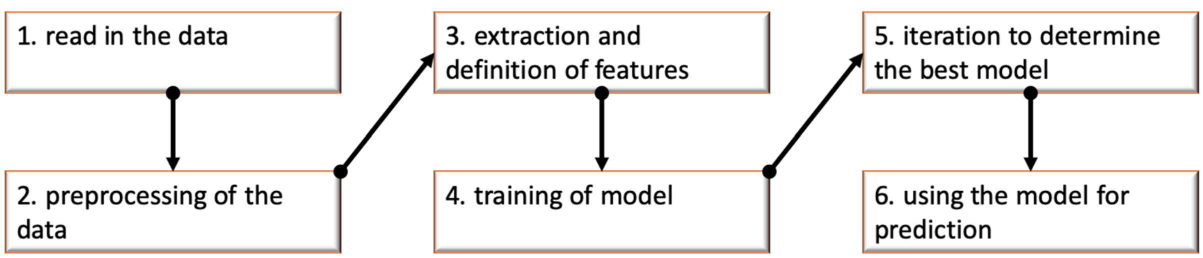

Fig. 2 Flowchart of generally valid scheme of steps involved in a successful ML project, from (Kraus and Drass 2020a)

3.2.1 0. Requirements for the AI Tool

Before presenting the entire process of the ML project, it was necessary to define the specifications for the ML tool in detail with the customer. The specifications of the tool were given by Kuraray Europe GmbH and are summarized in the following list:

- Application of AI to predict the weighted sound insulation value.
- No restrictions in the input, so that any glass assembly can be predicted in terms of arbitrary glass thicknesses, interlayer types, interlayer thicknesses, laminated glasses, double and triple insulating glasses with different thicknesses of the cavity as well as different gas fillings.
- Prediction accuracy for the sound insulation value of ± 1 dB.
- Deployment of the model on a website.

In particular, the prediction accuracy of 1 dB was a challenge in the prediction, which is described in more detail below.

3.2.2 1. Read the data

In step 1, existing data from the customer in form of an excel spreadsheet are compiled and brought in a form that an AI/ML model can access it and the learning algorithm is able to train on the data. This step may take minutes to months in dependence of the problem and the data structure of the respective environment, especially when digitizing existing older data from paper. It is advisable to consider standardization protocols for this step in order to guarantee data consistency within a company. It is important to note that the predictive power and accuracy of any data-driven model is based on the accuracy and quality of the input data.

The data structure for the present ML project was available as so-called structured data in tabular form. *Structured* data is information, which has a pre-defined data model (Frochte 2019; O'Leary 2013; Rusu et al. 2013), i.e. the location of each part of the data as well as the content is exactly know. Here, only the thickness of the glass, the interlayer type and thickness as well as the type of gas filling and the thickness of the gas filling were used as features. The experimentally determined sound insulation values, in this case R_W, STC and $OITC$, were used as labels or target values. The following list can be used to summaries the database and its features:

- **Laminated glass I**
  - Glass I-1 (2 - 15 mm)
  - Interlayer thickness I (0.38 - 4.56 mm)
  - Interlayer type I
  - Glass I-2 (2 - 15 mm)
    - Trosifol®Clear / Trosifol®UltraClear
    - Trosifol®SC Monolayer
    - Trosifol®SC Multilayer
    - Trosifol®ExtraStiff
    - SentryGlas®/ SentryGlas®Xtra™
  - Glass I-2
- **Cavity 1**
  - Cavity thickness I (1 - 30 mm)
  - Gas filling I (No Gas, Air, Argon, Krypton)
- **Laminated glass II**
  - Glass II-1 (2 - 15 mm)
  - Interlayer thickness II (0.38 - 4.56 mm)
  - Interlayer type II
    - Trosifol®Clear / Trosifol®UltraClear
    - Trosifol®SC Monolayer
    - Trosifol®SC Multilayer
    - Trosifol®ExtraStiff
    - SentryGlas®/ SentryGlas®Xtra™
  - Glass II-2 (2 - 15 mm)
- **Cavity II**
  - Cavity thickness II (1 - 30 mm)
  - Gas filling II (No Gas, Air, Argon, Krypton)
- **Laminated glass III**
  - Glass III-1 (2 - 15 mm)
  - Interlayer thickness III (0.38 - 4.56 mm)
  - Interlayer type III
    - Trosifol®Clear / Trosifol®UltraClear
    - Trosifol®SC Monolayer
    - Trosifol®SC Multilayer
    - Trosifol®ExtraStiff
    - SentryGlas®/ SentryGlas®Xtra™
  - Glass III-2 (2 - 15 mm)

With this basic structure of the data, all following steps in this project were carried out. The total number of entries in the database is about 1000 values and was provided by the customer without any preparation or cleaning.

3.2.3 2. Pre-processing of the data

After the data has been read into Python, the actual work of developing the ML tool by preparing the data began. Using programmed routines, the data was checked for missing values or incorrect entries. The results of the analysis showed no errors or statistically significant outliers in the database. Since the database is smaller than 1000 entries, this step of cleaning up the data was done automatically. A code snippet to check if there are zero entries in the database can be found in Appendix 1.

The result of listing 1 was that there are no null entries in the entire database, so that no further adjustments are necessary here. Common approaches in data pre-processing for dealing with zero values would be, for example, to delete them or to fill them up with the column averages (Frochte 2019).

In a further pre-processing step, the categorical variables (interlayer type and gas type) are dealt with. For the problem at hand, three strategies were considered:

- Deleting categorical variables
- Label encoding
- One-hot encoding

Both the implementation and the effect that deleting categorical variables can have on the result is easy to imagine and in some cases even leads to improved

Table 1 Comparison of handling categorical variables (deleting categorical variables, label encoding or one-hot-encoding) with respect to the mean average error of predicting the weighted sound insulation value

	Deletion	Label encoding	One-hot-encoding
MAE [dB]	2.57	2.51	**2.16**

Bold indicates shows that one-hot encoding leads to the best results

prediction accuracy. In comparison, label encoding and one-hot encoding are more complex than simply deleting columns of categorical variables in the spreadsheet. Label encoding means basically that the target labels will be encoded with values between 0 and n_classes − 1, while a one-hot encoder encodes each category as binary value. A one-hot is a group of entries among which the legal combinations of values are only those with a single high (1) entry and all the others low (0). The input to this transformer should be an array-like of integers or strings, denoting the values taken on by categorical features. The features are encoded using a one-hot (aka“one-of-K”or“dummy”) encoding scheme. This creates a binary column for each category and returns a sparse matrix or dense array (depending on the sparse parameter). By default, the encoder derives the categories based on the unique values in each feature (Rodríguez et al. 2018). This encoding is needed for feeding categorical data to many estimators. An example of one-hot encoding is shown in Fig. 3. The graph illustrates the procedure in relation to the categorical variables of different interlayer types.

In order to check which strategy of processing categorical variables is best, a Python function is implemented that makes their accuracy in their prediction comparable (see Appendix 1).

If we now run the function for the deletion of the categorical variables (see Appendix 1), label encoding (see Appendix 1) and one-hot encoding (see Appendix 1), we get the following results for the three approaches in terms of the Mean Average Error (MAE) (see Table 1):

For reasons of clarity, the Python codes for all three approaches were placed in the appendix. The final result shows that the best performance was achieved for the one-hot-encoding approach. Thus, in a final preprocessing step for the analyzed, categorical variables such as the interlayer type and the type of gas filling (No Gas, Air, Argon, Krypton) were converted into discrete numerical values using the one-hot-encoding method. In general it can be assumed for most ML algorithms that label-encoding is better suited for ordinal categories (data can be ranked e.g. pain intensity) and one-hot-encoding for nominal categories (data cannot be ranked e.g. gender). If nominal categories are label-encoded, the model might misinterpret the data based on the assigned order of numbers, assuming it to be ordinal (Géron 2019).

3.2.4 3. Extraction and definition of features

The extraction and definition of features is an essential step in ML projects. For the present ML project, the selection of features was of minor interest, since a weak feature correlation could be observed in the data analyzing the so-called feature correlation matrix.

A feature correlation matrix is a table showing correlation coefficients between features. Each cell in the table shows the correlation between two variables. In correlation matrices like the one shown below, it is important to keep the following points in mind:

- Each cell in the grid represents the value of the correlation coefficient between two variables.
- The value at position (a, b) represents the correlation coefficient between features at row a and column b. This will be equal to the value at position (b, a)
- All diagonal elements are 1. Since diagonal elements represent the correlation of each variable with itself, it will always be equal to 1.
- The axes ticks denote the feature each of them represents.
- A large positive value (near to 1.0) indicates a strong positive correlation, i.e., if the value of one of the variables increases, the value of the other variable increases as well.
- A value near to 0 (both positive or negative) indicates the absence of any correlation between the two variables, and hence those variables are independent of each other.
- Each cell in the matrix below is also represented by shades of a color. Here blue shades of the color indicate smaller values while red shades correspond to larger values (near to 1). This scale is given with the help of a color-bar on the right side of the plot.

As can be clearly seen in Fig. 4, strong correlations between the individual features only occur in the last

Fig. 3 Example of one-hot-encoding related to different interlayer types (Trosifol® Clear UltraClear, Trosifol® SC Monolayer, Trosifol® SC Multilayer, Trosifol® Extra Stiff and SentryGlas®) produced by Kuraray Europe GmbH

ID	Interlayer
1	PVB
2	SC Mono
3	SC Multi
4	ExtraStiff
5	Sentry

ID	PVB	SC Mono	SC Multi	Extrastiff	Sentry
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

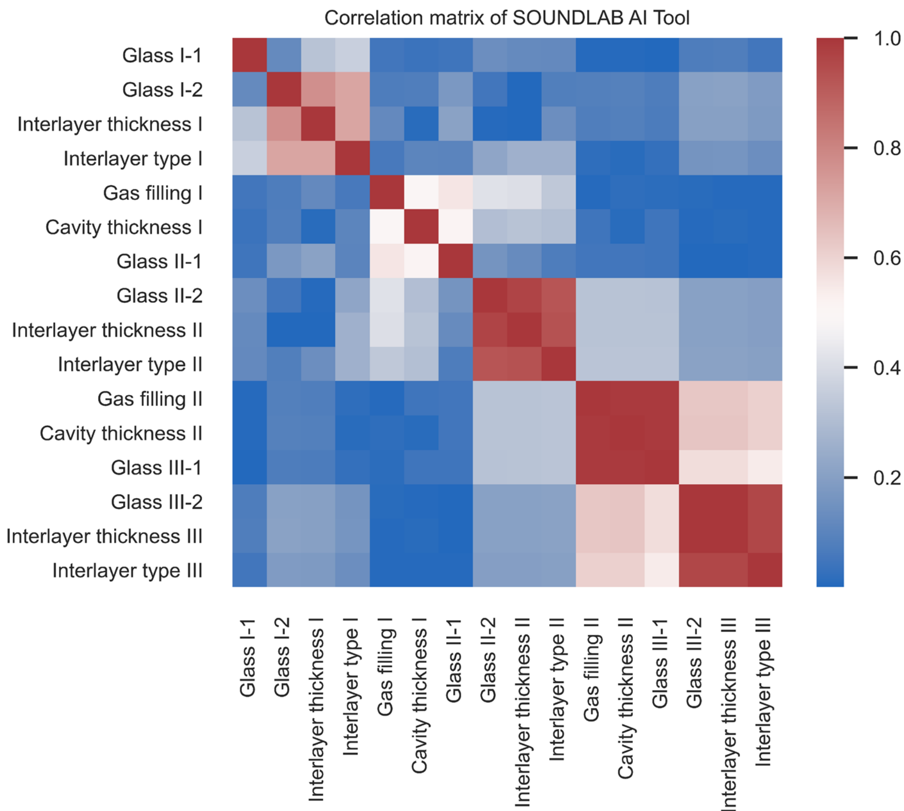

Fig. 4 Feature correlation matrix of all features for ML tool

elements from position II of the glass structure. In the literature, an unnecessary feature is often deleted from the dataset at a correlation value of 0.9. However, since only a maximum of 16 features are available in this study, all features are used in the following for training the ML algorithm.

3.2.5 4. Training of model

This essential part of a ML project is about the training of one or more ML algorithms. For this purpose, two steps have to be done in advance. On the one hand, suitable ML algorithms have to be selected which are suitable for the regression task for the AI-based prediction of the assessed sound insulation value for different glass structures. For the ML tool, three types of regression models have been trained to evaluate the general performance:

- Linear Regressor
- Decision Tree Regressor
- Random Forest Regressor

Linear regression models are not described in detail in the following, as it can be assumed that the basic principles are known. A decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. A Random Forest is a classification and regression procedure consisting of several uncorrelated decision trees. All decision trees are grown under a certain type of randomization during the learning process. Random decision forests correct for decision trees' habit of overfitting to their training set (Goulet 2020).

Before evaluating the performance of the above-mentioned ML algorithms, data has to be split into a training and a validation set accordingly to the classical holdout method (Raschka 2018). Additionally, a separate test dataset must be kept, which is neither used for training nor for validation of the algorithm. This separate test dataset is the latest experimental results provided by Kuraray Europe GmbH in this publication.

After pre-processing and visualization of the data, different ML models are evaluated. The main objective is to obtain a robust ML model that is able to generalize the extracted knowledge well to data that was not used during training (Mitchell 1997).

This means that at the end of the training process, the final model should correctly predict the training data, while at the same time it should also be able to generalize well to previously unseen data. Poor generalization can be characterized by overtraining or overfitting. This can be used to describe a model that produces very accurate results for the training samples but is unable to produce good results for data that is too different from the training data (Goodfellow et al. 2016). These two crucial demands (good prediction on training data as well as good generalization abilities) are conflicting and also known as the *Bias and Variance dilemma* (Bishop 2006). In order to judge how well a ML model performs on data, there exist several types of methods for evaluation (i.e. validation) (Raschka 2018):

- holdout validation
- k-fold cross validation
- stratified K-fold cross validation
- leave-one-out cross validation (LOOCV)

The simplest method for validation is holdout validation, in which the dataset is split into training and validation data over a fixed percentage value. Using the holdout method is perfectly acceptable for model evaluation when working with relatively large sample sizes (Raschka 2018).

In order to decide on the performance of the three ML models in this section, the entire dataset was divided into a 60/40 split and the training and validation of the models was carried out subsequently. It should be noted here again that a separate test dataset is also kept for the final test of the algorithm's quality, which is used neither for training nor for validation and hyperparameter tuning. Based on these preliminary studies, the prediction error plot and the cumulative distribution function (CDF) are presented for each model. At this point, it should be noted that this step is merely a preselection of a suitable ML algorithm and that more detailed explanations of the algorithm and its hyperparameters follow in the section on hyperparameter tuning.

Returning to the results of Fig. 5, a prediction error plot and the corresponding CDF are shown with respect to the above-mentioned three ML algorithms. A prediction error plot shows the actual target values from the dataset against the predicted values generated by the present ML model. This allows to obtain the amount of variance together with a notion of potential bias in the model. Data scientists can diagnose regression models using this plot by comparing against the 45 degree line, where the prediction exactly matches the model. In this plot, we only show the performance of the model on the validation dataset. This means that the applied ML models has not seen this data until after training. Being more precisely, in Fig. 5 one can see blue dots which represent the actual validation data, which corresponds to the real measurements of the sound insulation value of different glass assemblies. The x-axis indicated as"y2 represents the real, physical values of the sound insulation value, whereas the y-axis indicated as"$\hat{y}$"describes the predicted values of the sound insulation value. In the case that the model corresponds exactly to the measurements, this results in the so-called identity line, which runs at an angle of 45 degree. The "best fit"line, on the other hand, is the result of our ML model for the approximation of the assessed insulation value.

In contrast, a cumulative distribution function describes the cumulative probability of any given function below, above or between two points. Similar to a frequency table that counts the accumulated frequency

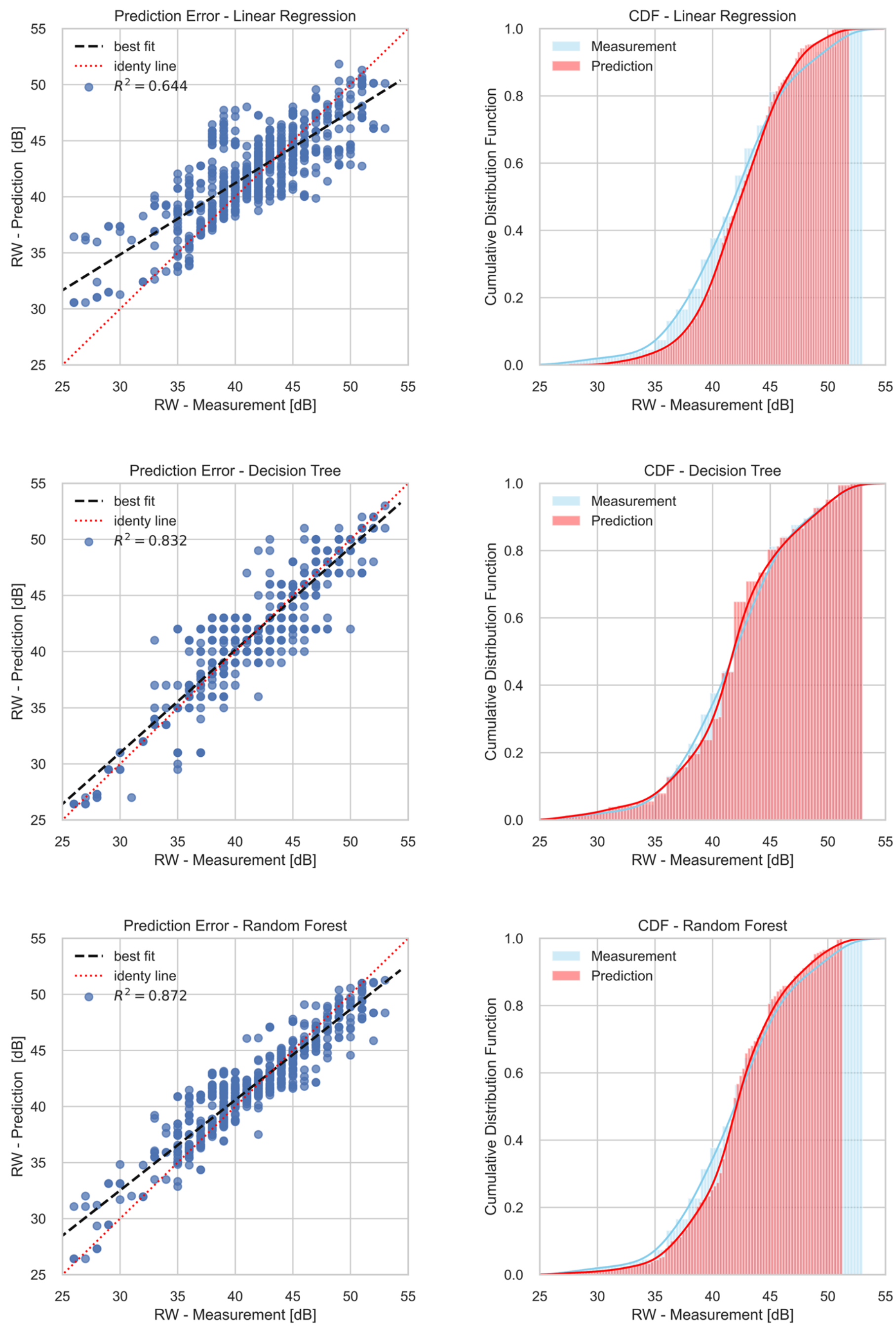

Fig. 5 Prediction error plot with a direct comparison of measurement data with the approximation by the best fit line evaluated only for the validation dataset and illustration of the cumulative distribution function for different ML models

of an occurrence up to a certain value, the CDF tracks the cumulative probabilities up to a certain threshold. Besides finding the probability of a random variable below or between two points, one can find the probability of a random distribution above a particular threshold. The latter is a technique called the complementary cumulative distribution function, or tail distribution, and as is quite useful in hypothesis testing. Finally, the CDF can be used to visualize the distribution between measured data and predicted data, as it is here the case.

As can be seen in Fig. 5, the linear regression model is unsuitable to perform the complex task of predicting the weighted sound insulation value, which is evident from the poor R^2 value. The R^2 score that specifies the goodness of fit of the underlying regression model to the validation data. In contrast, the decision tree model and the random forest regressor give very good results even without hyperparameter tuning. However, since decision trees generally tend to overfit, which also indicates a lack of generalization capability, the random forest regressor is used for the following hyperparameter tuning. If one looks at the CDF, the very good result by the tree models is also evident here, whereas the regression model clearly underestimates reality, especially in the range of small to medium R_W values, which is on the unsafe side from a practical construction point of view.

3.2.6 5. Iteration to determine the best model

In order to decide from the abundance of ML models the most suitable one and to fine-tune the inherent model parameters (hyperparameter tuning), the Random Forest ML model was selected in the previous section.

To optimize the inherent model parameters of the random forest regressor, a cross validation (CV) approach was used to get the best model for predicting the sound insulation value of arbitrary glass assemblies. Additionally, CV reduces the risk of so-called over- and underfitting as well by allowing a better estimation of the generalisation ability of a model. CV is a validation technique for assessing how the results of a statistical analysis will generalize to an independent dataset (Raschka 2018). The k-fold cross validation for example has a single parameter k, which refers to the number of groups into which a given data sample is divided. As such, the procedure is often referred to as k-fold cross validation, where the k is replaced with the specific choice to form the concrete name (e.g. $k = 5$ becomes a 5-fold cross-validation as schematically depicted in Fig. 6).

In the present study, a 5-fold cross validation approach with a split defined from the previous section of 60/40% of training and validation data was applied. The resulting hyperparameters of the random forest regressor read:

- max_depth = 11
- n_estimators = 35
- max_features = 15
- bootstrap = False
- min_samples_split = 4

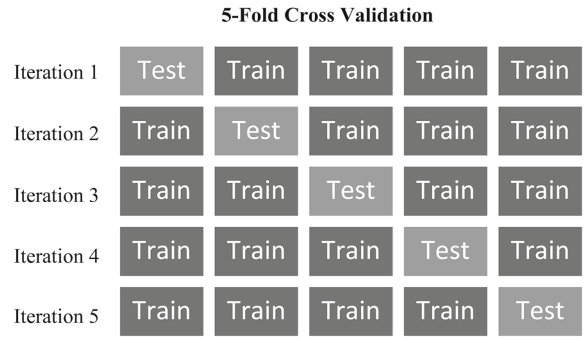

Fig. 6 Example of a 5-fold cross validation, from (Kraus and Drass 2020a)

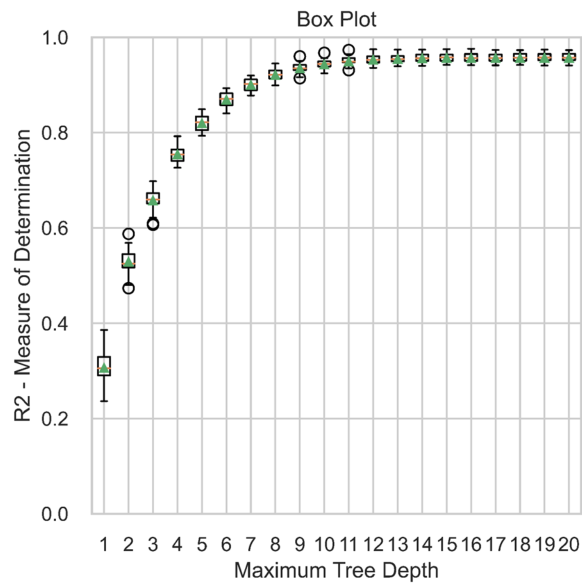

Fig. 7 Box Plot of Random Forest maximum tree depth vs. R2 accuracy

max_depth represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data. Choosing a high number for max_depth, the ML model will most likely overfit. Therefore, this parameter must be chosen careful.

To examine the maximum depth of the trees, a loop was programmed and the quality of the prediction was analysed as a function of max_depth. As can be seen in Fig. 7, there is no performance gain from a depth of 11, so this parameter was chosen accordingly to avoid overfitting.

n_estimators stands for the number of trees in the forest. The higher the number of trees, the better the

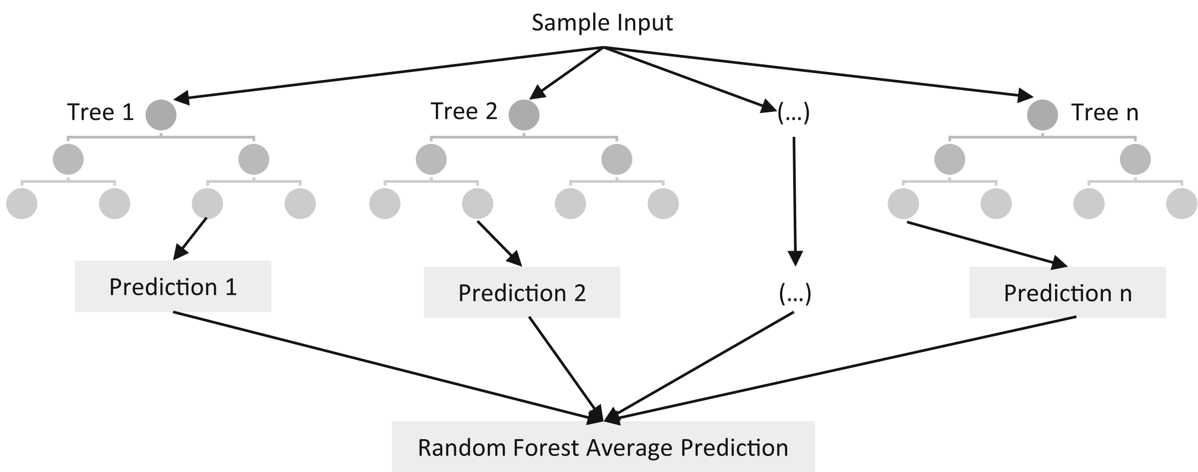

Fig. 8 Schematic presentation a random forest and its decision trees

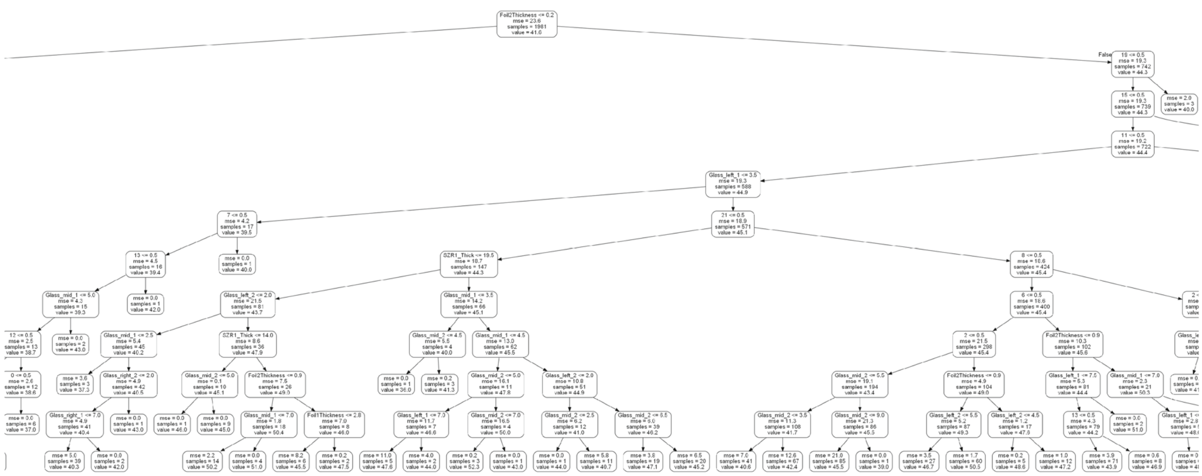

Fig. 9 Illustration of a part of the present Random forest ML model with focus on an aperture of decision tree number five

data can be learned. However, adding a large number of trees can slow down the learning process considerably, which is why we perform a parameter search to find the optimal point. According to the procedure described above, a loop was programmed to check the performance for different values of the. As a result, n_estimators = 35 was chosen due to the fact of no performance gain within the training set. For reasons of conciseness, this plot is not shown.

In order to show what the architecture of the selected random forest looks like, the Scikit Learn implementation of the random forest offers the possibility to draw a so-called dendrogram. A dendrogram is a diagram that represents a tree. Since a random forest naturally consists of individual trees, a tree must be selected when it is illustrated.

Figure 8 shows a schematic representation of a random forest and its decision trees, which run in parallel without any interaction. The final prediction is an average of all predictions of each decision tree. The individual levels of the singular decision trees are defined by the parameter max_depth. Figure 9 shows an example of decision tree five. It is easy to see that the depth has exactly 11 levels, as defined in the previous section. On the basis of this representation, it is possible to understand the prediction of the ML model, although the representation of such deep random forests or deci-

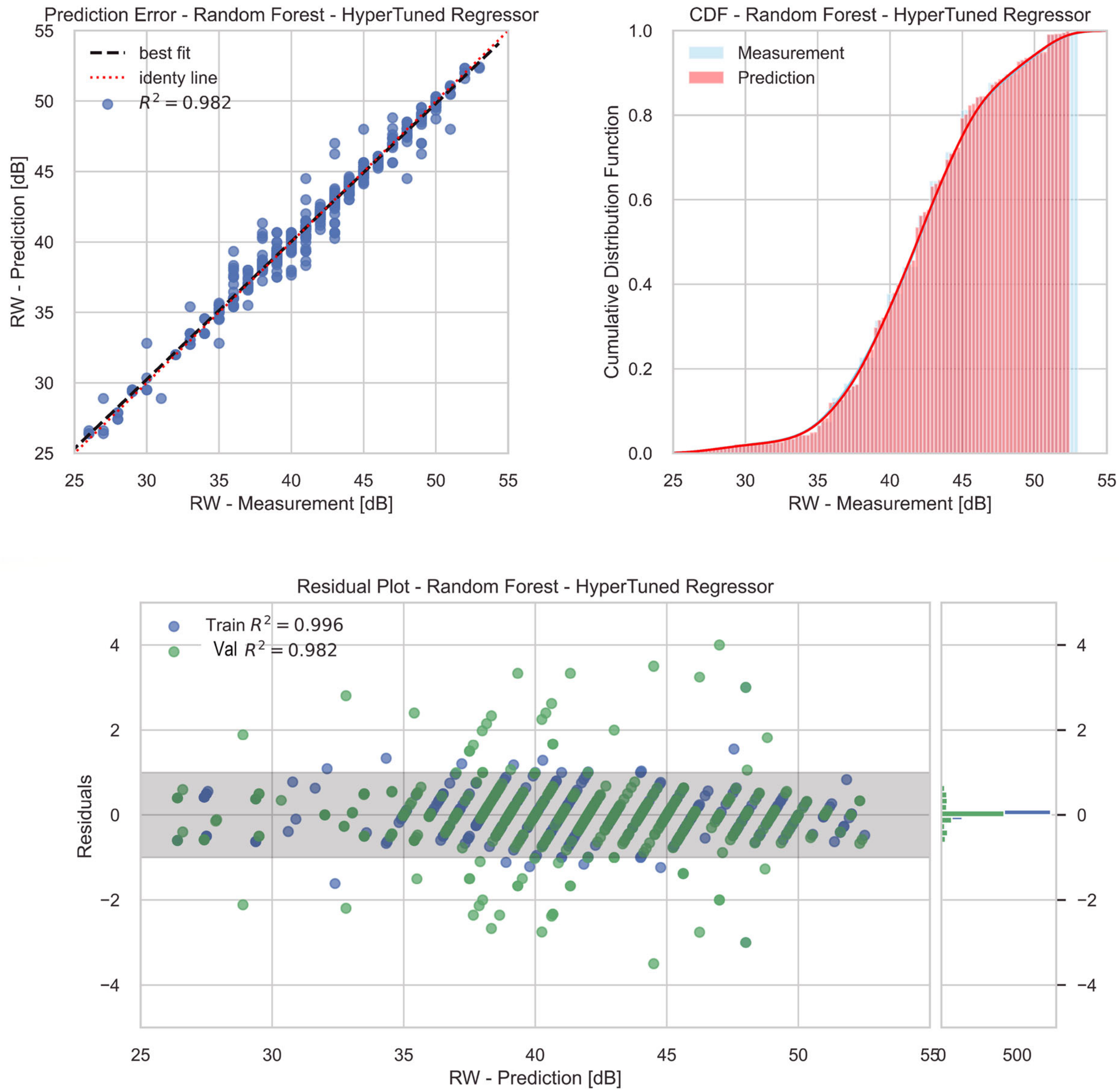

Fig. 10 Prediction error plot of ML Tool with a direct comparison of measurement data with the approximation by the best fit line evaluated only for the validation dataset, illustration of the cumulative distribution function and representation of the residual plot for the hyperparameter-tuned random forest regressor

sion trees is not always easy to understand or easy to represent due to their size.

This final model is now retrained and validated. For this purpose, the prediction error plot and the cumulative distribution function are shown again.

In addition, the so-called residual plot is added in the following evaluation. A residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. A special feature of this plot is the separate display of the residuals for training and validation set as well as the display of a grey bar representing the customer's specification of ±1dB for the weighted sound reduction index R_W. Ideally, the majority of the residuals should lie in this grey area to meet the client's requirements.

Looking at the accuracy of the present and final ML tool, one can see an excellence performance in the prediction error plot. Looking at Fig. 10, it can be concluded that we have reached a $R^2 = 0.996$ for training data and $R^2 = 0.982$ for validation data. This shows that the present ML model is very well suited for predicting sound insulation values. Looking at the cumulative distribution function, an excellent agreement between the measured data and the predicted values by the ML tool can be observed.

Finally, focusing on the residual plot described above, both training and validation data lie in the confidence band of $\pm$1dB in most cases, so that the customer's specification is met.

3.2.7 6. Model Accuracy for Test-Set

In this section, the hyperparameter-tuned model from the previous section will be used to make prediction on unseen data (test data), which correspond to a real test dataset which has not been used for training or validation and hypertuning of the ML model. A new dataset on special glass assemblies has been provided by Kuraray Europe GmbH, which is used as test dataset. Hence, 20 different glazing set-ups have been additionally experimentally analysed with regard to the sound insulation value. In the following, the model accuracy is presented as prediction error plot. As can be seen, the model provides very good predictions for data that were not the basis of the training of the algorithm which manifest itself with by $R^2 = 0.947$ (see Fig. 11).

3.2.8 7. Using the model for prediction

All necessary steps have now been taken to train a predictive ML model to predict the weighted sound insulation value. The last step is to provide the ML model to the customer. For this purpose, a web app application was programmed in Python using Flask and mod_wsgi for server deployment (Ronacher 2021; Dumpleton 2021D). The tool allows the input of any desired glass assembly while intercepting invalid inputs. The user interface is shown in Fig. 12. The web app consists of only a single input and output page, making the AI tool as user-friendly and easy to use as possible. Looking at the left input mask in Fig. 12, any glass structure with different thicknesses and interlayers can be entered here. The glass thicknesses, however, are limited to the common glass thicknesses of 2, 3, 4, 5,

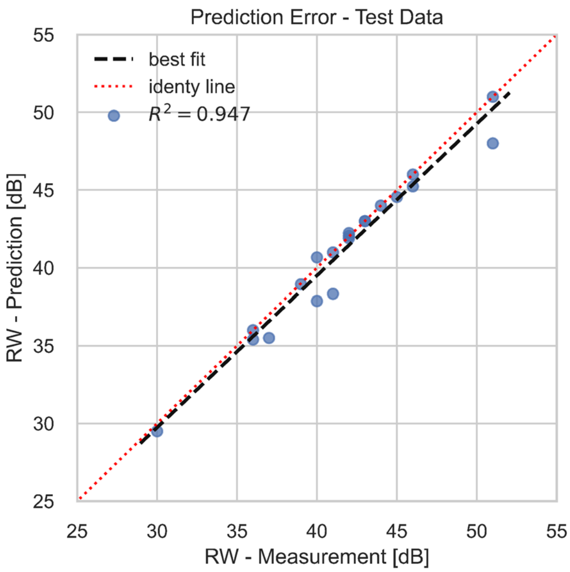

Fig. 11 Prediction error plot of ML tool with a direct comparison of measurement test data with the approximation by the best fit line evaluated only for the test dataset for the final hyperparameter-tuned random forest regressor

6, 8, 10, 12, 15 and 19 mm. The products Trosifol®-Clear / Trosifol®UltraClear, Trosifol®SC Monolayer, Trosifol®SC Multilayer, Trosifol®ExtraStiff and SentryGlas®/ SentryGlas®Xtra$^{\text{TM}}$ from Kuraray Europe GmbH can be used as interlayers. A maximum of one triple insulating glass can be entered into the tool. The structure of the input is as follows:

- Laminated glass 1
- Cavity 1
- Laminated glass 2
- Cavity 2
- Laminated glass 3
- Cavity 3

Once the desired input is made, the user presses "Predict"and the ML algorithm calculates the evaluated sound insulation values for R_W, STC and $OITC$. In addition to the prediction, an uncertainty quantification has been programmed, which gives an estimation of how reliable / uncertain the prediction is. In addition to these outputs, the tool indicates whether an experimental data point is available in the database. The output mask of the tool is shown in Fig. 12 (right). In order to understand the inputs of the server and to output the correct values mod_wsgi (module Web Server Gateway Interface) is used to translate the Hypertext Transfer Protocol (HTTP) to Flask-compatible values. Flask

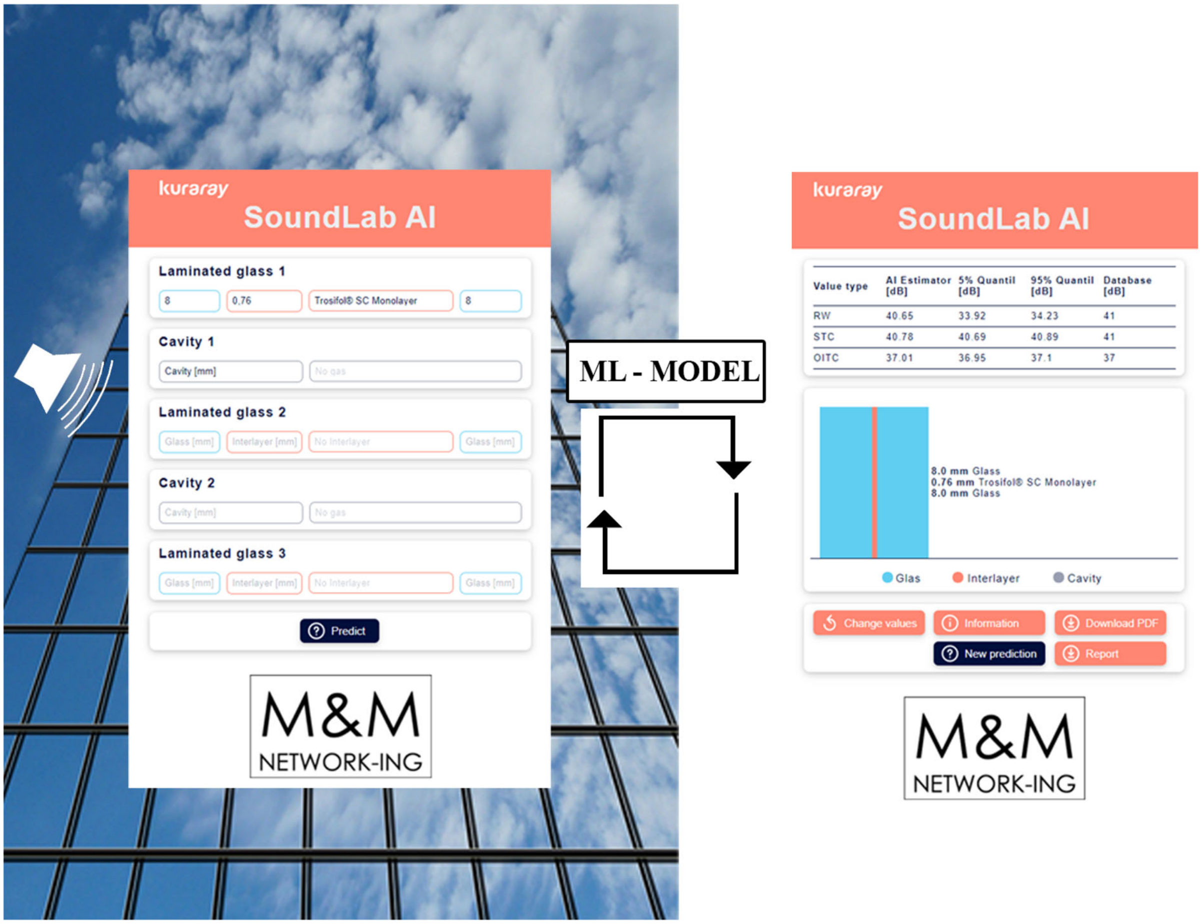

Fig. 12 Visualization of SOUNDLAB AI: input and output masks

then triggers the corresponding function and returns new values back to mod_wsgi as appropriate. This means that Python code can be run on pretty much any server without making any major adjustments, while the website is still displayed using standard HTML and CSS documents. The web app was launched in 2021 and is available at the following link: https://soundcalculator.trosifol.com/

3.3 Limitations of the ML model

3.3.1 Glass assembly

The ML model described here can predict the weighted sound reduction index for any glass structures of different glass thicknesses, different interlayer thicknesses, symmetrical and asymmetrical structures of insulating glass. However, the prediction only concerns the glass, so that the frame or support structures are excluded.

3.3.2 Glass thickness effect and damping

In the manuscript, it is described that the glass thickness effect and damping of the system has an influence on the sound insulation value. However, the model does not consider such effects explicitly but implicitly as a result of the measurements and the accurate approximation of the measurement data using machine learning. This approach is primarily a purely data-driven approach, but can be further developed by defining physical effects as losses in the loss function and thus extending the purely data-driven approach into a so-called physics-informed approach. However, a physics-informed approach has not yet been implemented, so it is not presented here and reference is made to the literature (Moseley et al. 2020; Kraus and Drass 2020a).

4 Summary and discussion

In this case study, the ML project between M&M Network-Ing and Kuraray Europe GmbH was briefly presented. This ML project for developing the so-called SoundLab AI Tool was described, which is an AI-based prediction tool to infer sound insulation values of arbitrary glass assemblies. The idea was to predict the weighted sound insulation value for glazing systems, as this value can only be determined by very complex numerical simulations or expensive experiments in classical approaches. The demonstrated ML tool was trained on structured data in a supervised learning scheme. The data were obtained within an extensive experimental program. The accuracy in the prediction error plot shows a very high predictive ability, which could be proven by a $R^2 = 0.996$ value for training data and $R^2 = 0.982$ for validation data. In addition, the ML model was also checked for previously unexploited test data. For the test dataset consisting of 20 entries, which was neither used for training nor for hypertuning / validation, a coefficient of determination of $R^2 = 0.947$ was achieved, which is a very good result.

The tool developed is therefore is a suitable method for making predictions of the sound insulation of any glass assembly quickly, cost-effectively and efficiently, which is a great advantage for the planning architects and engineers, especially in early project phases. The software-tool will be available online and provided by Kuraray Europe GmbH on the homepage for a broad audience.

Declarations

Conflict of Interests The authors declare the involvement in the development and depolyment of SoundLab AI.

Appendix A code-snippet: count number of missing values

```
#Number of missing values in data-set X

import numpy as np

missing_val_count_by_column = (X.isnull().sum())
print(missing_val_count_by_column[missing_val_count_by_column > 0])
```

Listing 1 Code - Missing values

Appendix B code-snippet: processing of categorical variables

```
from sklearn.ensemble import RandomForestRegressor from
sklearn.metrics import mean_absolute_error

train_X, val_X, train_y, val_y = train_test_split(X, y_RW,
train_size=0.8, test_size=0.2, random_state=0)

# Function for comparing different approaches def
score_dataset(train_X, val_X, train_y, val_y): model =
RandomForestRegressor(n_estimators=100, random_state=0)
model.fit(train_X, train_y) preds = model.predict(val_X) return
mean_absolute_error(val_y, preds)
```

Listing 2 Code - Function for comparing different approaches

Appendix C code-snippet: deleting categorical variables and performance test

```
drop_X_train = train_X.select_dtypes(exclude=['object'])
drop_X_valid = val_X.select_dtypes(exclude=['object'])

reduced_X_train = drop_X_train.dropna(axis='columns')
reduced_X_valid = drop_X_valid.dropna(axis='columns')

print("MAE from Approach 1 (Drop categorical variables and NaN):")
print(score_dataset(reduced_X_train, reduced_X_valid, train_y,
val_y))

# All categorical columns object_cols = [col for col in
train_X.columns if train_X[col].dtype == "object"]

# Columns that can be safely label encoded good_label_cols = [col
```

```
for col in object_cols if set(
    train_X[col]) == set(val_X[col])
    ]

# Problematic columns that will be
    dropped from the dataset
bad_label_cols = list(set(
    object_cols)-set(good_label_cols
    ))

print('Categorical columns that
    will be label encoded:',
good_label_cols) print('\
    nCategorical columns that will
    be dropped
from the dataset:', bad_label_cols)
```

Listing 3 Code - Function for deletion of categorical variables

Appendix D code-snippet: label encoding and performance test

```
from sklearn.preprocessing import
    LabelEncoder

# Drop categorical columns that
    will not be encoded
    label_X_train =
train_X.drop(bad_label_cols, axis
    =1) label_X_valid =
val_X.drop(bad_label_cols, axis=1)

# Apply label encoder label_encoder
     = LabelEncoder() for col in
set(good_label_cols): label_X_train
    [col] =
label_encoder.fit_transform(train_X
    [col]) label_X_valid[col] =
label_encoder.transform(val_X[col])

print("MAE from Approach 2 (Label
    Encoding):")
print(score_dataset(label_X_train,
    label_X_valid, train_y, val_y))
```

Listing 4 Code - Label Encoding

Appendix E code-snippet: one-hot-encoding and performance test

```
# Columns that will be one-hot
    encoded low_cardinality_cols = [
    col
for col in object_cols if train_X[
    col].nunique() < 10]

# Columns that will be dropped from
     the dataset
high_cardinality_cols =
list(set(object_cols)-set(
    low_cardinality_cols))

print('Categorical columns that
    will be one-hot encoded:',
low_cardinality_cols) print('\
    nCategorical columns that will
    be
dropped from the dataset:',
    high_cardinality_cols)

from sklearn.preprocessing import
    OneHotEncoder

# Use as many lines of code as you
    need!

OH_encoder = OneHotEncoder(
    handle_unknown='ignore', sparse=
    False)
OH_X_train =
pd.DataFrame(OH_encoder.
    fit_transform(train_X[
    low_cardinality_cols]))
OH_X_valid =
pd.DataFrame(OH_encoder.transform(
    val_X[low_cardinality_cols])) #
Your code here

# One-hot encoding removed index;
    put it back OH_X_train.index =
train_X.index OH_X_valid.index =
    val_X.index

# Remove categorical columns (will
    replace with one-hot encoding)
num_X_train = train_X.drop(
    object_cols, axis=1) num_X_valid
     =
val_X.drop(object_cols, axis=1)

# Add one-hot encoded columns to
    numerical features OH_X_train =
pd.concat([num_X_train, OH_X_train
    ], axis=1) OH_X_valid =
pd.concat([num_X_valid, OH_X_valid
    ], axis=1)

print("MAE from Approach 3 (One-Hot
     Encoding):")
print(score_dataset(OH_X_train,
    OH_X_valid, train_y, val_y))
```

Listing 5 Code - One-Hot-Encoding

References

Bishop, CM.: (2006) Pattern Recognition and Machine Learning, 1st edn. Information science and statistics, Springer

Bzdok, D., Altman, N., Krzywinski, M.: (2018) Points of significance: statistics versus machine learning

Chen, F., Huang, Y., DeGanyar, T.: (2019) Sound insulation of monolithic or laminated single-and double-glazing panels. Prediction methods for sound insulations Aachen

Chowdhary, K.: (2020) Introducing artificial intelligence. In: Fundamentals of Artificial Intelligence, Springer, pp 1–23

DeGanyar, T., Chen, F., Huang, Y.: (2019) Sound insulation of fenestration systems: a comprehensive web-based simulation program and validation

Dumpleton, G.: (2021) mod_wsgi documentation. https://modwsgi.readthedocs.io/en/master/, accessed: 2021-03-26

Frochte, J.: (2019) Maschinelles Lernen: Grundlagen und Algorithmen in Python. Carl Hanser Verlag GmbH Co KG

Goodfellow, I., Bengio, Y., Courville, A.: (2016) Deep learning. MIT press

Goulet, JA.: (2020) Probabilistic Machine Learning for Civil Engineers. MIT Press

Géron ,A.: (2019) Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly UK Ltd

Kraus, M., Drass, M.: (2020a) Artificial intelligence for structural glass engineering applications–overview, case studies and future potentials. Glass Structures & Engineering pp 1–39

Kraus M, Drass M (2020b) Artificial intelligence in civil engineering-background, status quo and potential applications. BAUINGENIEUR 95(10), 369–378

Kraus, MA.: (2019) Machine learning techniques for the material parameter identification of laminated glass in the intact and post-fracture state. PhD thesis, Universität der Bundeswehr München

Kraus, MA., Taras, A.: (2020) Physik-informierte künstliche intelligenz zur berechnung und bemessung im stahlbau. Stahlbau 89(10), 824–832, https://doi.org/10.1002/stab.202000074,https://onlinelibrary.wiley.com/doi/abs/10.1002/stab.202000074, https://onlinelibrary.wiley.com/doi/pdf/10.1002/stab.202000074

Kurra S (2012) Comparison of the models predicting sound insulation values of multilayered building elements. Applied Acoustics 73:575–589, DOI: https://doi.org/10.1016/j.apacoust.2011.11.008

Lee, J.H., Shin, J., Realff, M.J.: Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering **114**, 111–121 (2018)

Mitchell, TM.: (1997) Machine Learning, 1st edn. McGraw-Hill series in computer science, McGraw-Hill

Moseley, B., Markham, A., Nissen-Meyer, T.: (2020) Solving the wave equation with physics-informed deep learning.2006.11894

Murphy, KP.: (2012) Machine learning: a probabilistic perspective. MIT press

O'Leary DE (2013) Artificial intelligence and big data. IEEE intelligent systems 28(2):96–99

Raschka, S.: (2018) Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808

Rebala, G., Ravi, A., Churiwala, S.: (2019) An Introduction to Machine Learning. Springer

Rodríguez P, Bautista MA, Gonzalez J, Escalera S (2018) Beyond one-hot encoding: Lower dimensional target embedding. Image and Vision Computing 75:21–31

Ronacher, A.: (2021) Flask documentation. https://flask.palletsprojects.com/en/1.1.x/, accessed: 2021-03-26

Rusu, O., Halcu, I., Grigoriu, O., Neculoiu, G., Sandulescu, V., Marinescu, M., Marinescu, V.: (2013) Converting unstructured and semi-structured data into knowledge. In: 2013 11th RoEduNet International Conference, IEEE, pp 1–4