BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(1), 2026, Article number: e156771
DOI: 10.24425/bpasts.2025.156771

Predictive algorithms for determining blowing pressure in a pipe organ

Damian T. WĘGRZYN[1]*, **Piotr WRZECIONO**[2], and **Alicja A. WIECZORKOWSKA**[1]

[1] Polish-Japanese Academy of Information Technology, Warsaw, Poland
[2] Acoustic Measurements Laboratory, Warsaw University of Life Sciences, Warsaw, Poland

Abstract. The reconstruction of a pipe organ involves determining the blowing pressure. The lack of information about the pressure value significantly prolongs the process of instrument restoration. In addition, it may even result in irreversible damage to the pipes, as the adjustment of the sound parameters that depend on the pressure requires changing the physical structure of the pipes. In this paper, we provide a methodology for determining the blowing pressure in a pipe organ. We also present a formula describing the air pressure in the pipe foot, depending only on the height of the pipe's cut-up and the fundamental frequency. We apply machine learning to determine the blowing pressure, based on the parameters of only a percentage of pipes. Moreover, we use generative artificial intelligence, which achieves outstanding prediction accuracy. We conclude that the height of the cut-up and the fundamental frequency allow determining the blowing pressure. The more pipes, the higher the accuracy, but even 10% of pipes can be sufficient.

Keywords: pipe organ; blowing pressure; foot pressure; reconstruction; machine learning; generative artificial intelligence.

1. INTRODUCTION

Fire can destroy valuable works, as happened in Notre-Dame de Paris and in St. Elizabeth's Church in Wrocław. Warfare, robbery, or even incompetent repairs may also destroy pipe organs or the original features of the instrument. The historical pipe organ in Wrocław burned completely, and rebuilding it took years. The reconstruction work may employ IT methods [1, 2], which were also used in this case. The data for the reconstruction were obtained by analyzing historical sources and other preserved instruments built by the same organbuilder [3]. However, such data are not always available; sometimes the instrument must be restored from its picture only [4].

The pipe organ is one of the oldest musical instruments. At the end of the 4th century, CE, bellows were invented to supply organs with pressurized air called wind. The usage of a big pipe set requires a mechanism to regulate the pressure supplied to the pipes. Initially, water was used to stabilize air pressure. Ensuring the stability of pressure in the windchest of a pipe organ is a fundamental issue, as it guarantees the stability of the parameters of the generated sound. Today, this is achieved using pressure regulators, which control the airflow in particular pipes [5]. Weights placed on the bellows adjust the set pressure value. Occasionally, there are large pipe organs that use not just one, but several air supply systems for different sets of ranks [6]. Air supply systems provide constant air pressure in each pipe, which ensures even volume throughout the instrument.

*e-mail: damian@wegrzyn.info

Manuscript submitted 2025-05-05, revised 2025-08-26, initially accepted for publication 2025-10-19, published in January 2026.

A pipe organ has a fixed relative air pressure, i.e., pressure relative to the respective atmospheric pressure. It is measured using a water-filled U-shaped tube (water manometer), with one end open to the atmosphere, and the measured pressure applied to the other end. The manometer reading shows the difference between the tops of the water column on both sides, caused by raising the initial water level by the measured pressure of the windchest. The difference in water level is usually measured in millimeters of water gauge (mm H_2O), or in inches, and shows the pressure difference between the pressure in the windchest and the environment. This unit is commonly used in blowing pressure measuring, and the Pascal unit, from the International System of Units, is generally not used; 1 mm $H_2O = 9.80665$ Pa. Relative pressure is a permanent attribute of the organ because all the pipes of the organ are adjusted to the appropriate pressure value, which affects the basic sound parameters of the pipes. The pipe organ consists of flue (labial) and reed (lingual) pipes, and the same low-pressure value is used for both. Very rare are the special reed pipes, for which the pressure is 255 mm water column [7]. Flue pipes are always present in pipe organs and constitute the vast majority of all pipes. The exception is the regal, which consists solely of reed pipes, but this is a very rare instrument.

The value of air pressure in the pipe organ is essential not only during its construction, but also in the reconstruction process. It is usually impossible to determine its value based on the preserved elements of the wind pressure system. In such a case, it is necessary to change the structure of the (usually antique) pipes, e.g., by cutting them or deforming the pipe's mouth. Thus, incorrect selection of the pressure by the organbuilder may lead to the destruction of the instrument due to irreversible damage to the pipes.

In this work, we extend the previous research on the determination of blowing pressure based on the parameters of the pipes [8]. Our main goal is to protect the reconstructed instruments and facilitate the work of organbuilders in the reconstruction process. Restoring the blowing pressure is a time-consuming process, and our methodology helps reduce the time and costs involved. This is important because many damaged instruments require reconstruction, e.g., in the case of fire or warfare.

So far, there are no methods to reproduce the pressure value used in a damaged pipe organ, based on the pipes alone. Air pressure values vary by region and era. Baroque pipe organ usually has lower pressure than Romantic ones, and French pipe organ differs from German ones in terms of air pressure and the way of the pipe voicing [5]. The pressure may also depend on the style and period of construction of the instrument, the material used to build the pipes, the size and design of the instrument, the construction of the windchest, the way of playing at that time, the nature of the place, etc. However, the pipe sound generation is a physical phenomenon, in which the style of organ construction, era, or geography is irrelevant. The edge tone, one of the two sources of sound generated in an organ pipe, is related to the cut-up. The second sound source is at the top of the pipe. In the case of a properly voiced pipe, both of these sound sources have to be in tune.

Reproducing the blowing pressure is much easier when all pipes remain, as the bellows are then usually retained as well. However, in most cases, some or even most of the pipes are missing in the reconstructed pipe organs, and then the pressure cannot be calculated, analytically, numerically, or using computational fluid dynamics (CFD) methods. The pressure in the bellows is determined by the load placed on it. In this way, the pressure in the bellows is stabilized, almost independently of the degree of filling. Wedge bellows behave differently because the load on the bellows depends on the opening angle, but they are rare. Since we are dealing with a proportional-integral-derivative (PID) controller stabilization system, with the blower as the flow source, and the bellows applied to stabilize the pressure, CFD methods are not applicable, as we are not simulating the flow. Instead, in the process of organ reconstruction, we are dealing with an opposite situation than in the time of building the pipe organ, when other parameters of the instrument were adjusted to the blowing pressure. This reverse approach, where we determine the blowing pressure based on the physical characteristics of the pipes, is a form of reverse engineering.

The motivation for our research was to verify whether it is possible to determine the blowing pressure value based on an incomplete set of pipes. This boils down to establishing the relationship between the parameters of the remained pipes and the pressure of a pipe organ. Nowadays, machine learning (ML), including deep learning (DL), contributes significantly to the development of acoustics [9]. Unlike conventional acoustics and signal processing, ML relies on a data-driven approach. With sufficient training data, ML can uncover complex relationships between features and the predicted values, as well as interactions among the features themselves. By leveraging large datasets, ML can develop models that capture various acoustic phenomena, e.g., source separation, acoustic modeling, and timbre analysis [10, 11]. This is why we decided to use ML to find a solution to a problem that cannot be solved analytically.

Intuitively, this problem can be solved by training artificial neural networks (ANNs) on the recordings of pipe organ sounds, but they usually do not exist or cannot be obtained due to the lack of pressure information. The results obtained in our tests confirm that it is possible to overcome the problem of the lack of pipes and determine the pressure with high accuracy. ML and DL models trained can determine the pressure based on data from other pipes of the instrument. The main features from which the models infer the pressure value are the cut-up height of the labial pipes and the air pressure in the pipe's foot. We also give a formula for determining the pressure in the foot depending on its construction features.

A reconstruction process without basic data is difficult, and usually relies on already available methods or software [2]. In our case, this is not possible, so we created software for reproducing the basic parameters of the windchest, i.e., the blowing pressure.

The rest of the paper is organized as follows. Section 2 presents the methodology of the research, and Section 3 describes the data that were used for training and testing the models. Section 4 describes the input data processing, and Section 5 presents the obtained results. In Section 6, we compare our formula with other studies in the area of the analysis of flue pipes and blowing pressure, including a discussion of the limitations of our study. Section 7 concludes our work.

The main contribution of this research is the development of a methodology for determining the blowing pressure in a pipe organ. It empirically confirms the relationship between the physical features of the labial pipes and the blowing pressure, which allows the training of a model to determine the value of this pressure with high accuracy. The second contribution is a proposal for a formula describing the air pressure in the foot of the pipe. The next innovation is the incorporation of generative artificial intelligence (AI) to check what architectures it proposes for prediction, and to compare them with our results obtained using ML. An additional finding is the identification of key parameters influencing the blowing pressure. Additionally, we prepared a multi-threaded implementation of Weka regressors, which accelerated their training with various hyperparameters. This work also improves the pipe organ parameters reconstruction process and contributes to the area of pipe organ science in establishing the value of the pressure in the windchest, and paves the way for further research.

2. METHODOLOGY

Based on the so far good results of using ML in the field of examining the influence of the pipe's mouth parameters on the characteristics of the generated sound [12], we decided to investigate the relationship between the pipe parameters and the blowing pressure using ML, especially ANNs. The purpose of this study is to create a model that, after receiving selected pipe attributes at the input, will indicate the blowing pressure at the output. To train a model that works in various situations, a suf-

ficiently large set of input data is needed. Since the pressure value for pipe organs does not vary much and is usually in the range of 50–100 mm of water gauge [13], data from four instruments with different blowing pressures were used (66, 71, 74, and 94 mm H_2O). The same pipe stops (e.g., principal, octave, flute) repeat in various pipe organs. Materials from which organ pipes are made affect the timbre of the generated sound, i.e., the harmonics, but not its basic physical parameters, such as the fundamental frequency [14]. This allows data collection to be limited to only a few instruments, without the need to use data from multiple instruments.

Based on these data, we generated one million input datasets, where each input dataset simulates one damaged instrument. Each dataset contains data from one instrument only, representing the percentage of randomly selected pipes from all pipes available in this instrument. For 75% of pipes, one dataset contained an average of 180 samples, for 50% an average of 90 samples, for 30% an average of 55 samples, and for 10% an average of 20 samples. Creating all possible subsets of k pipes within a single instrument with n pipes would require generating all possible variations without repetition V_n^k, defined as

$$V_n^k = n!/(n-k)!, \tag{1}$$

where n is the number of all pipes and k is the number of pipes remaining in the damaged instrument. Usually, in a pipe organ, $n \gg 1000$. For example, if 50% of the pipes remain ($k = 500$, $n = 1000$), we would need about $3.3 \cdot 10^{1433}$ variations, and with 10% of the instrument remaining ($k = 100$, $n = 1000$), about $6 \cdot 10^{297}$ variations. Therefore, we have not conducted this research on all possible subsets, as such inputs are extremely big data. It is technically difficult to train a model on all variations due to the time required and/or memory limitations. On the other hand, there is no need to train the model on all possible subsets; thus, we limited training to one million input datasets. This value seems reasonable because it is large enough and does not require extensive random-access memory (RAM) resources.

2.1. Solutions to key issues encountered

Creating a million instruments to be used as input data causes problems. The first one is to ensure the uniqueness of randomly generated instruments, as obtaining unique sets of pipes within one instrument ensures that, after dividing these data into randomly chosen train and test sets, there is no overlap between them, and no train data are used in tests. We applied 5-fold cross-validation during training. Additionally, we also tested the trained models on another (fifth) instrument [15], which was not used in training nor evaluation of the models, i.e., data gathered from another pipe organ, with a different blowing pressure (80 mm H_2O).

This research uses a pseudo-random number generator to draw pipes to be included in datasets. To avoid, on the one hand, the generator falling into periodicity [16] and, on the other hand, possible repetitions of pipe sets, a mechanism of indexing pipes in sets was implemented. All pipes within one instrument are numbered sequentially. After generating the sets of randomly selected pipes, each set was assigned a binary number, with the number of digits equal to the number of pipes in this instrument. Each digit corresponds to the presence (1) or absence (0) of the corresponding pipe in the set. Thanks to this one-hot approach, a binary number representing each set was obtained, which allowed verifying the uniqueness of the sets. In the case of repetitions, new sets were generated, and the uniqueness of the sets was verified again. We assume that a pipe set is unique if at least 20% of its pipes differ from other sets. This simulates differences between real organs, as the same stops in pipe organs frequently occur; too small differences (e.g., just one different pipe) could lead to overtraining of ML algorithms.

The second issue is the problem of memory (space) complexity. To accelerate the model learning process, the optimal solution is to load all the training data of the model into the memory. As mentioned before, the data were processed to ensure uniqueness, which increased space complexity. To ensure smooth operation of our software, we carefully control RAM usage, see Subsection 4.1.

2.2. Programming language, libraries, and generative artificial intelligence

When choosing a programming language for our software, two popular languages with libraries for ML were taken into account: Java and Python. We chose the Java language for its better memory management. Firstly, we needed a statically typed language (with variable types assigned before using them) to accurately reserve the space for variables and optimize memory usage. Secondly, we needed a mechanism of immutable objects that ensures that the object remains permanent after its creation, to achieve secure and efficient memory management. In addition, Java provides efficient support for multithreading, which allows the parallel execution of different tasks.

The Application Programming Interface (API) of the Weka library, version 3.9.6 [17], was used in the software development process. We chose the Weka platform due to its constantly updated API and a large collection of implemented ML methods. In our research, the problem of regression is addressed, i.e., the output of our software is a real number. Therefore, all regression algorithms implemented in Weka were tested, in various configurations and with various hyperparameter settings, to obtain the best solution. Namely, we used the following algorithms: random forest (RandomForest), linear regression (SimpleLinearRegression), support vector machine for regression (SMOreg), sequential minimal optimization (SMO), radial basis function network (RBFNetwork), and single- and multilayer perceptron (MultilayerPerceptron). These algorithms were designed for batch or incremental learning.

Additionally, we used the Deeplearning4j library, version 1.0.0-M2.1 [18], and the Nd4j sub-module, which allows loading, executing, and retraining TensorFlow models. We used several popular ANN models for regression problems in various hyperparameter configurations. Namely, we used convolutional neural networks and recurrent neural networks, including long short-term memory (LSTM) and dense (DenseLayer) layers. The results obtained using these models were compared with the results obtained using ML algorithms from the Weka library, see Section 4.

For all ML algorithms applied in our work, we used default values of their hyperparameters, unless specified otherwise in Section 5.

In this work, we also decided to use generative AI: Gemini, version 1.5 Pro [19], and Llama, version 3.2 [20], which are large language models (LLMs). The LLMs we used were not fine-tuned, and the default settings of their generation parameters were applied. These chatbots were applied to see what generative AI has to offer to analyze relationships between attributes, suggest hyperparameters for various ANNs, and train and test selected ANN models. Generative AI models, even without specific training in predicting pipe organ blowing pressure, can create valuable neural network architectures based on the provided attributes. They focus on key input data, applying proven design principles to capture complex relationships between features. The proposed models can serve as rapid prototypes for further optimization and leverage interdisciplinary knowledge, offering a flexible and efficient approach to solving the problem.

We found that Gemini hallucinates when data is imported from external sources, i.e., publicly available data in CSV format, as described in Section 4.1. The imaginary models achieved high accuracy and other results as expected, which could confirm their effectiveness. However, an in-depth analysis of the models and dialogue with the Gemini chatbot revealed hallucinations, i.e., fabricated information. Additionally, the results were unstable, as they varied between different chats. Therefore, we discarded models built on external data and only analyzed results for input data limited to 1315 lines that did not cause hallucinations when using Gemini. We did not encounter similar problems using a local LLM, i.e., Llama. Furthermore, the results obtained by Llama are better than those obtained using Gemini.

3. OUR DATA

Based on the analysis of the related research, we selected a small set of input attributes, described in this section. This allowed us to avoid data redundancy and, as a result, overfitting when creating the models. In our selection of features, we focused on ensuring their high predictive values, which influences the construction of more effective models. We use four input attributes, the values of which were different for each pipe. These attributes are: cut-up height, fundamental frequency, airflow velocity, and air pressure in the pipe's foot. The input data used to train and test models represents either measured or calculated values. The output value is the blowing pressure p_w, measured in millimeters of water gauge; this value is predicted using ML and DL. The air pressure in the foot of the pipe is different from the blowing pressure. The blowing pressure is usually stable for the entire instrument or sometimes for a large group of pipes, while the pressure in the foot of the pipe is different for each pipe. In the case of incomplete instruments, when bellows and a significant percentage of pipes are missing, it is difficult to determine the blowing pressure, and the trial-and-error method may lead to the destruction of historical pipes.

Our data describe 191 pipes representing 21 stops from five complete instruments. The collected data are publicly available in The Diapason [15, 21–25], a journal dedicated to the organ, harpsichord, carillon, and church music. The data representing our input and output attributes are hard to obtain, as access to most pipes in instruments is usually almost impossible, because they are set tightly. Access for measurements is possible when disassembling the instrument for renovation, which happens rarely. In the literature, lip dimensions are usually not published together with blowing pressure; they are most often published separately. Such data cannot be used for the training of ML algorithms, as we need both input and output values. This is why we only have data for five instruments.

The first (measured) attribute is the height h of the pipe mouth's cut-up, in millimeters, see Fig. 1. The values of this attribute were measured by organbuilders and were published in The Diapason, as mentioned above.

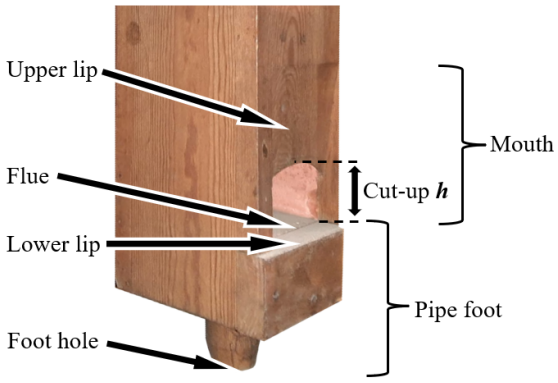

Fig. 1. Construction of a flue pipe with the height of the cut-up h indicated

The second (calculated) attribute is the fundamental frequency F_0 of the pipe's sound. It can be calculated in two ways. The first one is based on the wavelength of the sound

$$F_0 = c/\lambda, \tag{2}$$

where c is the speed of sound (m/s), and λ is the wavelength (m). In the case of instruments tuned in the equal temperament system, the second method to determine F_0 can be used. It is based on musical intervals and is calculated as the ratio of frequencies, which is constant and equal to $\sqrt[12]{2}$ for consecutive semitones in the twelve-tone equal-tempered scale, and tuned relative to a standard pitch A (around 440 Hz). We used this method to calculate the second attribute because it is faster than determining the wavelength for each pipe individually.

The third (calculated) input attribute is the velocity v of the airflow in a pipe, in meters per second. A constant value of the Strouhal number $S_t = 0.2$ was assumed for the calculations. The previous research published in [12] indicates that this value for labial pipes is constant when rounded to one decimal place, regardless of the type of pipe or its mouth. The airflow velocity v in a flue pipe is calculated as

$$v = F_0 \cdot h/S_t\,. \tag{3}$$

In addition, we assume that the pipe was properly voiced, which means that the pipe sounds as intended (without beats).

To achieve proper voicing, the edge tone produced in the cut-up must be of the same frequency as the resonator.

The fourth (calculated) input attribute is the air pressure p_p in the pipe's foot. As mentioned above, the value of p_p differs from the blowing pressure p_w and varies between pipes. When analyzing the outflow of air from the pipe's flue, this phenomenon can be considered as an adiabatic outflow of gas from a container with a higher pressure to an area with a lower pressure, with constant pressure in the container. In the case of a pipe organ, the pressure in the windchest and thus in the pipe foot is constant, as the roller valve and the bellows regulate the air supplied by the blower. Next, the wind duct supplies air to the windchest at the same pressure, see Fig. 2.

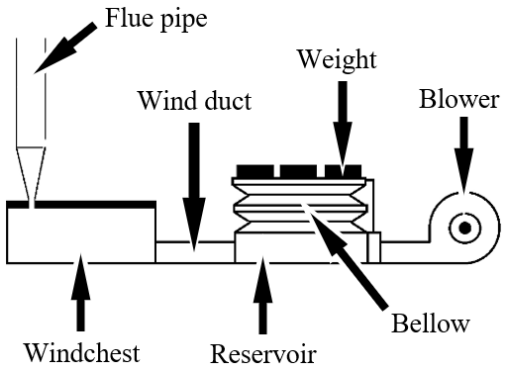

Fig. 2. Construction of the wind supply system of a pipe organ

Assuming an adiabatic outflow of an ideal gas without heat exchange and mechanical work from the container, the Saint Venant-Wantzel formula [26] for the velocity v of such outflow can be used

$$v = \sqrt{2\frac{\kappa}{\kappa-1}\cdot\frac{p_p}{\rho}\left(1-\left(\frac{p_e}{p_p}\right)^{(\kappa-1)/\kappa}\right)}, \tag{4}$$

where $\kappa = 1.401$ is the adiabatic exponent, p_p is the pressure in the pipe's foot (Pa), $p_e = 1013.25$ Pa is the pressure outside (ambient), and $\rho = 1.225$ kg/m^3 is the density of dry air at 15°C. In the situation of air intake by the blower from the same room where the windchest is located, the temperature is assumed to be constant. We can next transform (4), with the constants κ, p_e, ρ, as above, and use (3). As a result, we receive the following formula to determine the pressure p_p in the foot of a flue pipe, which depends on the pipe mouth cut-up height h and the fundamental frequency F_0 of the pipe's sound [8]

$$7.25\cdot\sqrt[1.401]{p_p} - p_p + 4.38\cdot F_0^2\cdot h^2 \approx 0, \tag{5}$$

where 7.25 and 4.38 are approximate values (rounded to two decimal places), resulting from transformations of (3) and (4). The approximation in (5) results from these approximations; for exact values, this formula is an equation.

As we can see, the fourth parameter (p_p) uses both the first attribute (h) and the second one (F_0), in a nonlinear equation. The pressure calculated using (5) does not consider losses in the airflow, and it indicates an absolute pressure. To determine the value of relative pressure p_{rel}, used in blowing pressure measurement, the ambient pressure p_e must be subtracted from the obtained pressure p_p, and the units converted to mm H_2O

$$p_{\text{rel}} = (p_p - p_e)/\xi, \tag{6}$$

where $\xi = 9.80665$ Pa/mm H_2O is the conversion factor applied to convert Pascals to millimeters of water gauge.

We use (5) to calculate the fourth input attribute, i.e., the air pressure in the foot of a pipe. This equation is a nonlinear, transcendental equation – it is not an algebraic equation because it contains the variable p_p under the root of the noninteger degree $\kappa = 1.401$. It is not possible to solve this type of equation analytically. Therefore, we attempted a geometric solution to achieve an intuitive insight into the structure of the solution. Based on the graphical solution, we can find that (5) has a unique solution. Unfortunately, this method proved to be inaccurate in determining the solution. Solving this equation requires the use of numerical methods that iteratively allow the solution to be determined with higher precision, especially for more complex values of fractional roots. In this work, we used the Newton-Raphson tangent method [27] as an iterative numerical method for finding the zeros of functions in a given range. It is often used in numerical analysis or optimization and is applied to nonlinear equations whose derivatives are easy to calculate. This method proved to be effective, and the solutions obtained were consistent with those obtained graphically.

4. DATA PROCESSING

We used the input data sets, prepared as shown in Section 3, to generate ML and DL models. We used the software we wrote for this research for this purpose. Additionally, we also used generative AI, as described in Subsection 4.2.

4.1. Programming ML and ANN models

The software for this research was written in the IntelliJ IDEA IDE, version 2023.3.4, using Java SE Development Kit, version 19.0.2, and Apache Maven, version 3.9.6. We used a Dell EMC PowerEdge R540 server with two Intel Xeon Silver 4110 2.10 GHz CPUs (16 physical cores) and 96 GB of ECC DDR4 2133 MHz RAM. Our software allows performing the following steps.

The first step is to set (by the user) the model options, and the number of pipes to be randomly selected in each draw from input data, to create the sets of pipes to be used in training and testing of ML algorithms. Next, data from CSV files are loaded into RAM. The file consists of four columns with input attributes and one column with an output attribute. All features are numeric, with two decimal places. The features (descriptors) of the pipes are: cut-up (mm), airflow velocity (m/s), fundamental frequency (Hz), and pressure in the pipe's foot (mm H_2O). The output data is the blowing pressure (mm H_2O).

The next step is to generate simulated sets of pipes, stored as two-dimensional arrays of double-precision floating-point numbers. Next, a procedure which ensures that each set is unique is executed, see Subsection 2.1. The unique sets that simulate pipe sets from different instruments are saved as two lists,

namely the list of the Instance class objects for ML algorithms in Weka, and the list of the DataSet class objects for DL models in Deeplearning4j. These lists of objects are randomly divided into a nonoverlapping train set (70%) and test set (30%). For memory efficiency, only the training and test lists are kept. Additionally, the train and test sets for both lists are normalized to the range [0, 1] after splitting into train and test sets, to avoid bias in the test data.

Next, regressors are trained on the prepared train data in a separate module. Additionally, we used five-fold cross-validation in training. All algorithms from the Weka library [17] and ANN models from Deeplearning4j [18], which are suitable for regression problems, were successively trained on the same data, see Subsection 2.2. The observed variance in the results within the folds for each model was negligible and did not affect the final ranking of the compared algorithms. Therefore, to maintain maximum clarity and to focus on the key performance differences between the models, we have chosen to present only the mean values of the performance.

The trained models are next tested on the test data. The following measures are calculated to evaluate the obtained results [28, 29]: the Pearson's correlation coefficient (r), mean absolute error (MAE), root-mean-square error (RMSE), relative absolute error (RAE), root relative square error (RRSE), mean absolute percentage error (MAPE), and acceptable error rate (AER), which is a version of the MAE adjusted to this research. The metrics used in the model evaluation are shown in (7)–(13)

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i}{\sqrt{\left[n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right] \cdot \left[n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right]}}, \tag{7}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}, \tag{8}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}}, \tag{9}$$

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} |x_i - \bar{x}|} \cdot 100\%, \tag{10}$$

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}} \cdot 100\%, \tag{11}$$

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left|\frac{x_i - y_i}{x_i}\right|, \tag{12}$$

$$\mathrm{AER} = t_3 \cdot 100\%/n, \tag{13}$$

where x_i is an actual value, y_i is the predicted value, $\bar{x}$ is an arithmetic mean of all actual values, n is the total number of predictions, and t_3 is a number of predictions such that $|y_i - x_i| \leq 3$. We used the AER measure because theadmissible error in the blowing pressure is ≤ 3 mm H_2O, which is caused by two reasons. Firstly, the measurement error made by the organbuilder is ≤ 1 mm H_2O. Secondly, the difference in pressure in the air supply system itself can vary because of differences in the height of the position of various components of the system relative to the ground. Assuming that the position difference between the bellow h_1 and the windchest h_2 is not greater than 3 m, the change in pressure between the bellow p_1 and the windchest p_2 may be circa 3 mm H_2O. This results from the Bernoulli's equation [30] for an ideal gas and adiabatic transformations

$$p_1 - p_2 = \rho\left(g\,(h_2 - h_1) + \left(\frac{v_2^2 - v_1^2}{2}\right)\right), \tag{14}$$

where $g \approx 9.81$ m/s^2 is gravitational acceleration, v_1 is the velocity of the air coming out of the bellows, and v_1 is the velocity of the air entering the windchest. Additionally, to ensure the reliability of the r correlation results, we applied the standard z-score measure, confirming the set absence of outliers (z-scores in the range from -1.34 to 1.97).

We encountered various technical problems when developing our software. We started with single-threaded processing of regressors, but with an increasing number of the generated sets of pipes, the time of model training increased to the order of hours. To speed up the calculations, each regressor we used has been rewritten to a multi-threaded version, using the Executor class and Lambda expressions in Java. Unfortunately, regressors from the Weka library in multi-threaded training require synchronization of model threads and the input data, which significantly increases the training time. Additionally, we had to deal with the RAM usage problem, to avoid the program stopping because of running out of memory, with the increase in the number of pipe sets. Therefore, RAM monitoring and memory cleaning via manual control of the garbage collector were used. Another problem was posed by a pseudo-random number generator applied in the process of generating the sets of pipes. In the case of a small number of pipes remaining in the set, we found repetitions in the generated sets. This confirms that it is necessary indeed to use a procedure that ensures the uniqueness of pipe sets, which we implemented as described in Subsection 2.1.

4.2. Application of generative AI

We used generative AI [19, 20] to suggest ANN architectures and compare its proposals with our results obtained from Weka and Deeplearning4j. We provided the chatbot commands with one prompt. In the case of Gemini, its length could not be longer than 1316 lines: 1315 lines of data and 1 line of instructions. Llama, however, was tested locally, on the server described in Subsection 4.1. The prompt for both LLMs[1] consists of:

- The command to analyze the data provided below.

[1] https://github.com/damianwegrzyn/LLMs_prompt

- Information that the data is numeric, stored in five columns separated by commas, with the first four columns storing input attributes, and the fifth one the output value.
- The command to propose three of the most adequate ANN architectures and their hyperparameters, based on the analysis of the provided data.
- The command to train the proposed ANN models on 70% of randomly selected data rows, which should be normalized and scaled before training.
- The command to evaluate models on the remaining 30% of data rows, which should be normalized and scaled before testing.
- The command to print numerical values of model evaluation measures (all accurate to four decimal places): r, MAE, RMSE, in percentage: RAE, RRSE, MAPE, and the AER calculated as $AER = t_3/n$
- Only for Gemini: input data in the form of 1315 lines (i.e., 1315 pipes).

The Gemini chatbot proposed ANN architectures and their hyperparameters presented in Table 1. Table 2 presents Llama's proposals based on the same prompt as Gemini, but for the full set of input data. The model evaluation measures are described in Subsection 4.1.

Table 1
The architectures and hyperparameters of the ANNs proposed by Gemini

ANN type	Architecture	Hyperparameters	Data preparation
Multilayer perceptron (MLP)	Input layer: 4 neurons Hidden layer 1: 16 neurons, rectified linear unit (ReLU) activation Hidden layer 2: 8 neurons, ReLU activation Output layer: 1 neuron, linear activation	Optimizer: Adam Learning rate: 0.001 No. of epochs: 100 Batch size: 32 Loss function: mean squared error (MSE)	The data was normalized and scaled using the standard MinMax scaler.
Recurrent neural network (RNN)	Input layer: 4 neurons Hidden layer 1: 16 neurons, ReLU activation Hidden layer 2: 8 neurons, ReLU activation Output layer: 1 neuron, linear activation	Optimizer: Adam Learning rate: 0.001 No. of epochs: 100 Batch size: 32 Loss function: MSE Dropout: 0.2 Bidirectional: True	The data was normalized and scaled using the standard MinMax scaler.
Convolutional neural network (CNN)	Input layer: 4 neurons Convolutional layer 1: 32 filters, size 3×3, ReLU activation Max pooling layer 1: Pool size 2×2 Convolutional layer 2: 32 filters, size 3×3, ReLU activation Max pooling layer 2: Pool size 2×2 Fully connected layer 1: 128 neurons, ReLU activation Output layer: 1 neuron, no activation	Optimizer: Adam Learning rate: 0.001 No. of epochs: 100 Batch size: 32 Loss function: MSE	The input data was normalized using standard scaling (z-score). The target value was scaled using a logarithm.

Table 2
The architectures and hyperparameters of the ANNs proposed by Llama

ANN type	Architecture	Hyperparameters	Data preparation
Deep neural decision forest (DNDF)	Input layer: 4 neurons Decision tree layer: 16 trees, max depth 5 Neural layer: 2 hidden layers, 16 neurons each, ReLU activation Output layer: 1 neuron	Optimizer: stochastic gradient descent Learning rate: 0.01 No. of epochs: 50 Batch size: 64 Loss function: MSE Early stopping: True	The data was normalized and scaled using the MinMax scaler.
Multilayer perceptron (MLP)	Input layer: 4 neurons Hidden layer 1: 32 neurons, ReLU activation Hidden layer 2: 16 neurons, ReLU activation Output layer: 1 neuron	Optimizer: Adam Learning rate: 0.001 No. of epochs: 100 Batch size: 32 Loss function: MSE	The data was normalized and standardized using z-score scaling.
Fully connected neural network (FCNN)	Input layer: 4 neurons Hidden layer 1: 64 neurons, Tanh activation Hidden layer 2: 32 neurons, Sigmoid activation Output layer: 1 neuron	Optimizer: root mean square propagation (RMSprop) Learning rate: 0.0005 No. of epochs: 100 Batch size: 128 Loss function: MSE	The data was scaled using the robust scaler. Outliers in target values were clipped.

5. RESULTS

We compared all trained models using the following evaluation metrics, listed in Subsection 4.1: r, MAE, RMSE, RAE, RRSE, MAPE, and AER. The most important metric in the assessment was the acceptable error rate (AER). The models that performed best were the random forest (RF) and the multilayer perceptron (MLP). We publish the best trained models for future research purposes[2], and present the dependence of their AER on the number of remaining pipes in the instrument in Fig. 3.

The perceptron and sequential minimal optimization for regression (SMOreg) algorithms also performed well. Table 3 presents evaluation results for the best three models, for four percentages of the pipes drawn into the pipe set (simulating the remained pipes in an instrument), namely: 75%, 50%, 30%, and 10% of the full set of pipes.

Considering the specificity of the data and the training target, we find the trained algorithms very useful. The value of Pearson's correlation coefficient of 0.97 for the RF with 75% of the pipes remaining indicates a strong correlation between the predictions and the actual values. A low mean value of absolute differences between model predictions and actual values, i.e., MAE, also indicates good performance and accurate predictions. The MAPE value, often used as a loss function in regression, was always less than 5.5%. This confirms that the

[2]https://github.com/damianwegrzyn/ML_models

Table 3
The evaluation of the top three ML algorithms for training on 75%, 50%, 30%, and 10% of all pipes

Remaining pipes	Rank	Model	AER [%]	r	MAE	RMSE	RAE [%]	RRSE [%]	MAPE [%]
75%	1	RF	96	0.97	0.80	1.84	13.39	22.40	1.04
	2	MLP	79	0.69	4.46	9.36	74.41	113.74	5.01
	3	Perceptron	74	0.54	4.99	8.36	83.32	101.56	5.23
50%	1	RF	93	0.95	0.92	2.43	15.41	31.05	1.27
	2	MLP	84	0.82	4.50	8.52	78.37	108.60	5.06
	3	SMOreg	72	0.54	3.34	6.91	55.62	88.30	4.07
30%	1	RF	85	0.86	2.05	4.20	33.29	50.29	2.62
	2	MLP	83	0.63	4.89	8.56	78.73	101.89	5.41
	3	RNN	68	0.44	4.50	8.49	72.75	99.00	5.49
10%	1	MLP	84	0.47	4.50	8.10	73.38	100.09	5.10
	2	SMOreg	76	0.09	4.49	9.27	71.18	111.89	4.99
	3	CNN	75	0.12	4.56	9.64	73.96	113.24	5.12

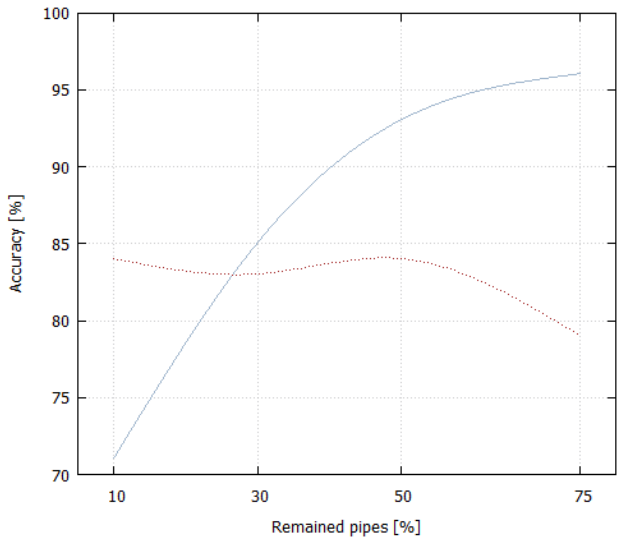

Fig. 3. The dependence of the accuracy of RF (grey solid line) and MLP (red dotted line) models on the number of remained pipes

output values predicted by these algorithms are close to actual values. The RMSE of 1.84 shows that large error values are rare; low RMSE indicates that predictions rarely differ much from actual values. RAE and RRSE allow comparing the quality of forecasts between different models and between input data of different scales. The obtained low values of these measures confirm the good quality of the obtained models. These results confirm that it is possible to predict the blowing pressure in a pipe organ using ML and the features described in Section 3.

The training was repeated several times (up to 50 repetitions for RF), and similar values of evaluation measures were obtained, which confirms the repeatability of the results. We built an RF with 200 decision trees with a maximum depth of 5 and the number of attributes to randomly investigate set to 4, but even an RF with only 10 trees yielded good results; thus, we present results for this small RF, which are even better than for 200 trees. For at least 30% of pipes remaining in the set, the RF was unbeatable and showed high resistance to outliers, but its quality dropped significantly for 10% of remaining pipes in the set.

In addition, we analyzed the importance of individual input attributes in our best RF model. We used the built-in functionality in Weka to determine the importance of attributes, which is calculated as mean decrease in impurity; the higher the score, the more important the feature is.

Figure 4 presents the importance of features in our best RF model. As we can see, all features contribute to predictions, and the height of the cut-up is the most important feature. This confirms the dependence of the blowing pressure on the features we used, and is consistent with organ building knowledge.

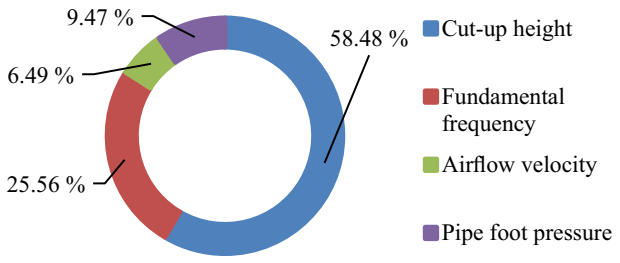

Fig. 4. The importance of individual input attributes in the best model RF for 75% of the remaining pipes

The perceptron and the MLP were adapted to solve the regression problem by changing the activation function to a linear function (no activation) and changing the cost function to mean squared error (MSE). Training was performed for 100 to 1000 epochs, with a step of 100 epochs, and with additional train-

ing for 1500 and 2000 epochs. Both models achieved the best results for the following hyperparameters: learning rate for the backpropagation algorithm equal to 0.3, momentum rate for the backpropagation algorithm equal to 0.2, training for 100 epochs, with a validation threshold equal to 20 (default value), the size of the validation set equal to 20, and the seed for the random number generator equal to 1. The MLP model consisted of three hidden layers.

The SMOreg model is also worth attention. It is a regression algorithm based on support vector machines that uses the sequential minimal optimization. This model is especially useful for nonlinear regression problems because it can automatically extract a data transformation function that allows the model to fit the data better. Models built using linear regression, i.e., linear regression and simple linear regression, yielded worse results, and their accuracy was low.

CNN and RNN models also yielded satisfactory results, especially for the lowest investigated percentage of remaining pipes: CNN for 10% and RNN with DenseLayer for 30% of remaining pipes. Thus, as the number of remaining pipes decreases, more complex ML algorithms are better at discovering the relationships within the input data.

We also applied generative AI to compare our results with the results obtained by Gemini AI (for data limited to 1315 lines). As shown in Table 4, these models achieved similar results for various types of ANNs.

Table 4

The evaluation of the ANN models proposed and created by Gemini on the 1315 pipes dataset

Rank	Model	AER [%]	r	MAE	RMSE	RAE [%]	RRSE [%]	MAPE [%]
1	MLP	95	0.99	2.78	4.23	5.94	10.13	3.87
2	RNN	94	0.99	2.75	3.93	5.24	7.85	2.14
3	CNN	92	0.95	2.13	3.02	6.22	9.16	5.43

The obtained model accuracies are high, but they were achieved for the limited input data set, which is in accordance with our expectations. It is worth noting that relying on limited input data for AI evaluation may not fully capture the model performance in diverse scenarios.

We also verified the ANN architectures proposed by Llama. The results obtained by Llama are presented in Table 5. The best accuracy was achieved by the deep neural decision forest (DNDF) [31] model: 97%, which is a better result than for the RF model. The DNDF is a decision tree model that guides representation learning in the initial layers of a CNN and replaces the traditional fully connected layers with a decision forest for the final prediction. This distinguishes DNDF from traditional neural networks. It is especially effective in the case of nonlinear relationships between input and output variables. The high performance of the RF and DNDF models indicates that the use of tree structures is an effective method for predicting blowing pressure. MLP and fully connected neural network (FCNN) also yield high accuracy, comparable to Gemini models.

Table 5

The evaluation of the ANN models proposed and created by Llama

Rank	Model	AER [%]	r	MAE	RMSE	RAE [%]	RRSE [%]	MAPE [%]
1	DNDF	97	0.99	2.02	2.34	15.54	14.34	1.38
2	MLP	95	0.98	3.49	4.06	16.49	16.73	2.17
3	FCNN	94	0.96	3.37	3.96	15.91	26.67	2.32

Such good results, obtained from both LLMs, open the possibility of further research toward model optimization. Modern AI learning support techniques, such as techniques embeddings AI, vector database, or retrieval augmented generation, can be used to train the ANN models on large datasets and yield even better results.

We also checked the performance of the best trained models, i.e., RF and MLP, using the data representing the fifth instrument [15] as a test set. This instrument was not used in training. The instrument comes from a different region (Germany) and era (baroque) than the pipe organs we used to train the models. The evaluation results are presented in Table 6. They confirm the usefulness of both models.

Table 6

The evaluation of RF and MLP models on the test instrument, not used in the training process

Remaining pipes	Rank	Model	AER [%]	r	MAE	RMSE	RAE [%]	RRSE [%]	MAPE [%]
75%	1	RF	94	0.95	0.97	2.02	14.02	28.20	1.19
	2	MLP	75	0.67	4.56	10.13	77.42	117.10	5.92
50%	1	RF	92	0.93	1.01	2.86	15.83	32.75	1.30
	2	MLP	82	0.77	4.62	8.64	80.00	114.21	5.53
30%	1	RF	84	0.86	2.11	4.32	37.09	59.04	2.84
	2	MLP	81	0.57	4.97	8.84	80.10	109.36	5.88
10%	1	MLP	83	0.47	4.50	8.74	75.43	104.83	5.27
	2	RF	70	0.24	5.00	10.22	67.04	91.28	4.71

6. RELATED RESEARCH AND DISCUSSION

Determining the blowing pressure in a pipe organ is not often discussed in the literature; works on sound pressure level (SPL) or acoustic pressure [32] and their measurement are more often found. Saenger [33] presents the method of SPL measurement in a labial resonator, and Rucz [34] shows the pressure distribution for various modes in the resonator. Smith [35] states that the sound generation process has two components: a standing wave and turbulence coming from the upper lip or tongue. Angster and Miklós [13] present the edge tone as a sound source in the pipe. Odya *et al.* [36] confirm that the second sound source is at the top of the flue pipe.

The influence of the cut-up height on the labial pipe voicing is confirmed by research and organ-building practice [37, 38]. Adachi *et al.* [39] prove that even a small change in the pipe's

mouth results in significant changes in the sound characteristics. The influence of the mouth on the sound is also seen in other flute instruments, such as the recorder or the transverse flute [40]. Each change of the upper lip results in a spectrum change, as studied by Stafura and Nagy [41]. Angster and Miklós [13] proved that shortening the upper lip reduces the amount of air flowing into the pipe; the decrease in air jet velocity results either from the increase in the distance of the upper lip from the flue (shortening of the upper lip) or from the decrease in pressure. The influence of lip parameters on the velocity of the air jet was confirmed by Fabre *et al*. [42]. Mallock [43] found that the fundamental frequency of the pipe's sound depends on the jet velocity. An increase in air velocity increases frequency, up to a certain limit. Our research results also show that the cut-up height is the most important attribute influencing the blowing pressure, which is consistent with the above-mentioned research.

The velocity of the air jet is usually determined from Bernoulli's equation [44,45]. Steenbrugge [46] and Steenbrugge and de Baets [47] published a formula for calculating the air jet flow velocity depending on the pressure in the pipe's foot. By modifying this formula, if the flow velocity is given, the value of the relative pressure in the pipe foot (in mm H_2O) can be determined

$$p_{\text{rel}} \approx \frac{\rho v^2}{19.6}. \tag{15}$$

The velocity of the jet coming from the flue was experimentally measured by Außerlechner *et al*. [48] and Elder [49]. The velocity values for similar pipes are very close to the values we calculated. The air jet velocity value was also experimentally measured in various pipes by Węgrzyn *et al*. [12]. Węgrzyn *et al*. [50] also confirmed the influence of the air jet velocity on the pipe's fundamental frequency.

The air pressure in the foot of the pipe depends mainly on the blowing pressure, but also on the geometric dimensions of the foot hole and the flue [13]. Fletcher and Rossing [7] stated that the foot hole (the air intake) of the pipe effectively controls the pressure exerted on the flue slit. Steenbrugge [46] experimentally confirmed the relationship between the air pressure in the foot of the pipe and the size of the foot hole. Angster *et al*. [51] proved that different types of windchests affect the pressure in the pipe foot, with the blowing pressure unchanged. Lunn [52] and Steenbrugge [46] experimentally confirmed that the pressure drop at the foot hole should be assumed. Due to frictional losses, the pressure in the foot of the pipe is less than the bellow pressure [52]. Kanda and Shimomukai [30] numerically proved that the pressure at the walls of the foot is lower than in the central core. This phenomenon even occurs for the Reynolds number lower than 5000. Außerlechner *et al*. [48] measured the pressure in the foot of the pipe using a pressure sensor. Pipes with similar geometrical features, used in their research, had pressure values similar to the values in our work, which confirms the correctness of the calculated pressure values.

The influence of changing the blowing pressure on the sound generated by pipes is well-known [6]. The change of the blowing pressure changes the amplitude of the generated sound, its pitch, and timbre [45, 53]. Steenbrugge and de Baets [47] confirmed the increase in sound frequency with increasing blowing pressure and vice versa. With the decrease in blowing pressure, the sound becomes darker and duller [47]. If pressure increases, a discontinuity of pressure at the mouth is observed, caused by the centrifugal force of the curvilinear flow [52].

A fixed value of blowing pressure allows for a strong and clear sound [36]. The blowing pressure cannot change significantly, and if the pressure is too high, overblowing, typical of wind instruments, occurs. Rucz *et al*. [45] analyzed the dynamic organ pipe invented by Zacharias. Due to the use of a blown open tongue with free reed, this pipe can be adjusted to blowing pressure between circa 5 and 110 mm H_2O. Tuned pipes, commonly used in the pipe organ, do not allow the blowing pressure to change significantly. Steenbrugge [54] also describes the feedback cycle operating regime for pipe blowing, which includes cut-up, blowing pressure, and the flue width. Therefore, tuned pipes are adjusted to a limited blowing pressure range.

6.1. Limitations of the study

The main limitation of this study is that our methodology can be applied only to pipe organs with bellows with a reservoir. It does not apply to wedge bellows, which are rare. Another key requirement is the presence of flue pipes. If only reed pipes survive, our method will not apply either. In addition, for the blowing pressure predictions to achieve acceptable accuracy, the remaining flue pipes should constitute at least 10% of all pipes. Moreover, it is also required to gather pipe attributes, namely cut-up height and fundamental frequency.

Due to the lack of dimensions of the pipe's mouth published together with the blowing pressure, our models were trained only on four pipe organs. From the point of view of organ building and the physics of sound generation, the attributes we selected allow us to generalize our methodology to other pipe organs.

The limitation of interpretation is the admissible error in the accuracy of the determined blowing pressure. It accepts a pressure error of 3 mm H_2O and a difference in the relative position of the windchest to the bellows of up to 3 m.

The implementation limitation of ML and ANN training was the restriction of input data to one million datasets, due to the available amount of 96 GB of the server RAM. Furthermore, in the case of using Gemini, the input data was limited to 1315 pipes due to the limitation of the input prompt in the case of this chatbot.

When analyzing the results of generative AI, it is worth noting the occurring hallucinations (Gemini and external data collection), i.e., a phenomenon in which an LLM generates information that is false, fabricated, or has no basis in its training data, but presents it confidently and convincingly. This carries the risk that the generated proposals for ANN architectures and the obtained evaluation values may be false. The second problem is the black box problem, i.e., the impossibility of fully understanding and explaining how chatbots achieve the presented results. Since an LLM internal processing is not based on human-interpretable rules but on complex patterns learned from data, it is difficult to identify subtle, hidden errors, bias contained in the training data, or the model limitations that could have influenced the final

results. However, we were aware of the aforementioned limitations. Thus, we verified the responses from the LLMs to avoid the impossibility of research reproducibility. The results from the LLMs are consistent with the outcomes from the evaluation of our trained models.

7. CONCLUSIONS

The issue of restoring blowing pressure in a pipe organ has so far been an unsolvable problem. In most cases, the restored organs are incomplete, with bellows and a high percentage of pipes missing. The proposed solution, based on ML and ANN models, yields high accuracy and confirms the possibility of determining the blowing pressure. Our methodology is currently the only alternative to the trial-and-error method, which may destroy the historical pipes, as there is no method to calculate the blowing pressure. We also propose a faster yet equally effective way of determining the blowing pressure using commonly available tools such as LLMs, which facilitates the practical application of our solution.

The most efficient ML model if the majority of pipes in the remaining instrument is the RF model, with an accuracy of 96%. In an extreme situation, when circa 90% of the pipes are missing, the MLP model is the most effective one, with an accuracy of 82%. Among the tested models that use the ANNs, the best predictor is the DNDF model, which achieved an accuracy of 97%. Other ANN models also achieved high results, exceeding 92%.

Equation (5) proposes a formula describing the air pressure in the labial pipe's foot depending only on its fundamental frequency and cut-up height, with no other variables as we used a constant Strouhal number, determined in the previous work for various mouth and pipe types [12]. We also confirmed the relationship between flue pipe attributes and blowing pressure. We found that the height of the cut-up is the most important feature. Fundamental frequency is also important, as it affects the proper voicing of a pipe. These two attributes suffice to determine the blowing pressure (the others can be calculated), and their importance is confirmed in RF.

In our training, we used a variety of organ pipes, i.e., different types, constructions, and mouths. They came from instruments with different blowing pressures. As mentioned in Section 1, the era or origin of the organ is not important, because we selected input attributes that characterize the physical properties of the sound, not its timbre. Thanks to this, we obtained diversity that allows for generalization of our methodology also to other pipe organs, not included in the training dataset. We confirmed this by testing our best ML models on the data from a pipe organ that was not used in training. It is worth noting that this pipe organ comes from a different era and region than those used in training.

The proposed methodology of determining blowing pressure is an important achievement, which applies state-of-the-art information technology in the broader process of renovation and reconstruction of historical objects, thus increasing financial efficiency and minimizing trial-and-error attempts. Sometimes, only illustrations of the pipes have been preserved, but the instrument has not survived [4]. If these graphics are of good quality, they may be used to restore the blowing pressure. It would significantly improve the instrument rebuilding process while maintaining the authenticity of the original sound. Our findings may help not only organ builders in the reconstruction of destroyed pipe organs, but also in determining the pressure in the windchest during new instrument building. Another area that may benefit is the analysis of fluid flows, especially in the problems of unknown value of constant fluid pressure. The next area of ??the application of our results is acoustic modeling. The ability to accurately determine the blowing pressure can aid in the creation of precise acoustic models that predict sound behavior depending on changes in the construction of the pipes or the windchest system.

In future work, we would like to investigate the influence of other organ pipe attributes (flue dimensions, scaling, i.e., ratio of pipe length to diameter, etc.) on blowing pressure. Moreover, the models for determining the blowing pressure can be optimized to be efficient even in the case of highly incomplete instruments. Designing a new neural network architecture or machine learning regression technique may yield better results than the commonly available models we used. Faster and more accurate algorithms would significantly shorten the process of reconstructing the instrument and facilitate the daily work of organbuilders.

ACKNOWLEDGEMENTS

Open access was founded by the Research Center of PJAIT, supported by the Ministry of Science and Higher Education in Poland.

REFERENCES

[1] R. Bork, "The Design Geometry of Notre-Dame in Paris," *J. Soc. Archit. Hist.*, vol. 81, no. 1, pp. 21–41, 2022, doi: 10.1525/jsah.2022.81.1.21.

[2] I. Trizio, E. Demetrescu, and D. Ferdani, Eds. *Digital Restoration and Virtual Reconstructions. Case Studies and Compared Experiences for Cultural Heritage*. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-15321-1.

[3] M. Szostak, "The rebirth of Michael Engler's treasure in Wrocław," *The Organ*, no. 400, pp. 4–19, May-Jul. 2022.

[4] P. Wrzeciono, "Pattern Recognition in Music on the Example of Reconstruction of Chest Organ from Kamień Pomorski," *Sensors*, vol. 21, no. 12, p. 4163, June 2021, doi: 10.3390/s21124163.

[5] D. Bush and R. Kassel, Eds. *The Organ: An Encyclopedia*. New York, NY, USA: Routledge, 2004, ch. 11, pp. 322–323, doi: 10.4324/9780203643914.

[6] J. Angster, P. Rucz, and A. Miklós, "Acoustics of organ pipes and future trends in the research," *Acoust. Today*, vol. 13, no. 1, pp. 10–18, 2017. [Online]. Available: https://acousticstoday.org/wp-content/uploads/2021/08/Acoustics-of-Organ-Pipes-and-Future-Trends-in-the-Research-Judit-Angster-1.pdf. [Accessed: May 9, 2025].

[7] N.H. Fletcher and T.D. Rossing. *The physics of musical instruments*. New York, NY, USA: Springer, 1998, doi: 10.1007/978-0-387-21603-4.

[8] D. Węgrzyn, P. Wrzeciono, and A. Wieczorkowska, "The Reconstruction of Blowing Pressure in Pipe Organ Using Machine Learning," in *Harnessing Opportunities: Reshaping ISD in the post-COVID-19 and Generative AI Era (ISD2024 Proceedings)*, B. Marcinkowski *et al.*, Eds. Gdańsk, Poland: University of Gdańsk, 2024, doi: 10.62036/ISD.2024.113.

[9] M.A. Roch, P. Gerstoft, B. Kostek, and Z.H. Michalopoulou, "How machine learning contributes to solve acoustical problems," *Acoust. Today*, vol. 17, no. 4, pp. 48–57, 2021, doi: 10.1121/AT.2021.17.4.48.

[10] M.J. Bianco *et al.*, "Machine learning in acoustics: Theory and applications," *J. Acoust. Soc. Am.*, vol. 146, no. 5, pp. 3590–3628, Nov. 2019, doi: 10.1121/1.5133944.

[11] J. Chen, K. Tatar, and V. Zappi, "A Deep Learning Framework for Musical Acoustics Simulations," *AIMC*, Aug. 2024. [Online]. Available: https://aimc2024.pubpub.org/pub/5cl1cvmy. [Accessed: May 9, 2025].

[12] D. Węgrzyn, P. Wrzeciono, and A. Wieczorkowska, "The Dependence of Flue Pipe Airflow Parameters on the Proximity of an Obstacle to the Pipe's Mouth," *Sensors,* vol. 22, no. 1, p. 10, Dec. 2022, doi: 10.3390/s22010010.

[13] J. Angster and A. Miklós, "Properties of the Sound of Flue Organ Pipes," in *Springer Handbook of Systematic Musicology*, R. Bader, Ed. Berlin – Heidelberg, Germany: Springer, 2018, pp. 141–155, doi: 10.1007/978-3-662-55004-5_8.

[14] A. Danihelová, A. Štafura, M. Čulík, and T. Gergež, "Influence of Wood and Thickness of Back Wall of Wooden Organ Pipe and Air Pressure in Windchest on Sound," *Appl. Sci.*, vol. 14, no. 17, p. 7897, Sep. 2024, doi: 10.3390/app14177897.

[15] M. McNeil, "The Sound of Gottfried Silbermann. Part 2," *The Diapason*, vol. 1357, no. 1, pp. 13–19, Jan. 2023.

[16] P. L'Ecuyer, "Random Number Generation," in *Handbook of Computational Statistics. Springer Handbooks of Computational Statistics*, J. Gentle, W. Härdle, Y. Mori, Eds., Heidelberg, Germany: Springer, 2012, doi: 10.1007/978-3-642-21551-3_3.

[17] Waikato Environment for Knowledge Analysis. "The Weka workbench," Weka 3 – Data Mining with Open Source Machine Learning Software in Java. [Online]. Available: https://ml.cms.waikato.ac.nz/weka. [Accessed: May 9, 2025].

[18] Eclipse Deeplearning4j Development Team. "Eclipse Deeplearning4j." GitHub. [Online]. Available: https://github.com/deeplearning4j. [Accessed: May 9, 2025].

[19] Google AI. "Gemini." Gemini. [Online]. Available: https://gemini.google.com/app. [Accessed: May 9, 2025].

[20] Meta AI. "Llama." Llama. [Online]. Available: https://www.llama.com. [Accessed: May 9, 2025].

[21] M. McNeil, "1863 E. & G.G. Hook Opus 322 Church of the Immaculate Conception Boston, Massachusetts Part 1," *The Diapason*, vol. 1292, no. 7, pp. 17–19, Jul. 2017.

[22] M. McNeil, "The 1864 William A. Johnson Opus 161, Piru Community United Methodist Church Piru, California, Part 1," *The Diapason*, vol. 1305, no. 8, pp. 16–20, Aug. 2018.

[23] M. McNeil, "The 1864 William A. Johnson Opus 161, Piru Community United Methodist Church Piru, California, Part 2," *The Diapason*, vol. 1306, no. 9, pp. 20–25, Sep. 2018.

[24] M. McNeil, "The 1864 William A. Johnson Opus 161, Piru Community United Methodist Church Piru, California, Part 3," *The Diapason*, vol. 1307, no. 10, pp. 26–28, Oct. 2018.

[25] R. Redman, "Louis Debierre Choir Organ, 1884," *The Diapason*, vol. 1298, no. 1, pp. 18–19, Jan. 2018.

[26] S.L. Arsenjev, I.B. Lozovitski, and Y.P. Sirik, "The Flowing System Gasdynamics, Part 3: Saint-Venant-Wantzel formula modern form," *ArXiv*, Feb. 2003, doi: 10.48550/arXiv.physics/0302038.

[27] S.C. Chapra and R.P. Canale, "The Newton-Raphson Method," in *Numerical Methods for Engineers*, 7th ed., S.C. Chapra and R.P. Canale. Eds,. New York, NY, USA: McGraw-Hill, 2002, pp. 151–156.

[28] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson Correlation Coefficient," in *Noise Reduction in Speech Processing. Springer Topics in Signal Processing 2*, I. Cohen, Y. Huang, J. Chen, and J. Benesty, Eds., Berlin – Heidelberg, Germany: Springer, 2009, doi: 10.1007/978-3-642-00296-0_5.

[29] M.V. Shcherbakov, A. Brebels, N.L. Shcherbakova, A.P. Tyukov, T.A. Janovsky, and V.A. Kamaev, "A survey of forecast error measures," *World Appl. Sci. J.*, vol. 24, pp. 171–176, 2013, doi: 10.5829/idosi.wasj.2013.24.itmies.80032.

[30] H. Kanda and K. Shimomukai, "Numerical study of pressure distribution in entrance pipe flow," *J. Complex.*, vol. 25, no. 3, pp. 253–267, 2009, doi: 10.1016/j.jco.2009.02.003.

[31] P. Kontschieder, M. Fiterau, A. Criminisi, and S.R. Bulo, "Deep Neural Decision Forests," in Proc. *IEEE Int. Conf. Comput. Vis. (ICCV),* 2015, pp. 1467–1475, doi: 10.1109/ICCV.2015.172.

[32] P. Bordoni, P. Odya, J. Kotus, and B. Kostek, "Sounding Mechanism of a Flue Organ Pipe - A Multi-Sensor Measurement Approach," *Sensors*, vol. 24, no. 6, pp. 1962, Mar. 2024, doi: 10.3390/s24061962.

[33] K.L. Saenger, "A pressure-based transfer matrix method and measurement technique for studying resonances in flutes and other open-input resonators," *J. Acoust. Soc. Am.*, vol. 147, no. 4, pp. 2556–2569, Apr. 2020, doi: 10.1121/10.0001102.

[34] P. Rucz, "Determination of organ pipes' acoustic parameters by means of numerical techniques," M.A. thesis, Budapest University of Technology and Economics, Hungary, 2009.

[35] H. Smith, "On the Physical Action taking place at the Mouth of Organ-pipes," *Nature*, vol. 10, pp. 161–163, Jul. 1874, doi: 10.1038/010161d0.

[36] P. Odya, J. Kotus, M. Szczodrak, and B. Kostek, "Sound intensity distribution around organ pipe," *Arch. Acoust.*, vol. 42, no. 1, pp. 13–22, Jul. 2017, doi: 10.1515/aoa-2017-0002.

[37] J. Liljencrants. "End Correction at a Flue Pipe Mouth." Johan Liljencrants on organs, pipes, air supply. [Online]. Available: https://fonema.liljencrantz.se/mouthcorr/mouthcorr.htm. [Accessed: May 9, 2025].

[38] S. Pitsch, P. Rucz, J. Angster, A. Miklós, and J. Kirschmann, "Scaling software for labial organ pipes," in Proc. *AIA-DAGA*, Merano, Mar. 2013.

[39] S. Adachi, J. Angster, and A. Miklós, "Numerical simulation of the flow in the flue organ pipe," *Fortschritte Der Akustik*, vol. 33, no. 1, pp. 225–226, 2007. [Online]. Available: https://pub.dega-akustik.de/DAGA_1999-2008/ data/articles/003154.pdf. [Accessed: May 9, 2025].

[40] S. Dequand, J.F. Willems, M. Leroux, R. Vullings, M. van Weert, C. Thieulot, and A. Hirschberg, "Simplified models of flue instruments: Influence of mouth geometry on the sound source," *J. Acoust. Soc. Am.*, vol. 113, no. 3, pp. 1724–1735, Feb. 2003, doi: 10.1121/1.1543929.

[41] A. Stafura and S. Nagy, "The upper lip as a voicing element of wooden organ pipes," *Akustika*, vol. 25, pp. 55–61, Mar. 2016.

[42] B. Fabre, J. Gilbert, and A. Hirschberg, "Modeling of wind instruments," in *Springer handbook of systematic musicology*, R. Bader, Ed., Berlin – Heidelberg, Germany: Springer, 2018, pp. 121–139, doi: 10.1007/978-3-662-55004-5_7.

[43] A. Mallock, "Note on the effect of wind pressure on the pitch of organ pipe," *Proc. R. Soc. Lond.*, vol. 95, no. 666, pp. 99–106, Oct. 1918, doi: 10.1098/rspa.1918.0051.

[44] N.H. Fletcher, "Sound production by organ flue pipes," *J. Acoust. Soc. Am.*, vol. 60, no. 4, pp. 926–936, Oct. 1976, doi: 10.1121/1.381174.

[45] P. Rucz, J. Angster, and A. Miklós, "Examination of a novel organ pipe construction with blown open tongue," *Fortschritte der Akustik*, vol. 42, pp. 1292–1295, 2016. [Online]. Available: https://pub.dega-akustik.de/DAGA_2016/data/articles/000309.pdf. [Accessed: May 9, 2025].

[46] D. Steenbrugge, "Fluid mechanical aspects of open-and closed-toe flue organ pipe voicing," *Int. J. Sustain. Construct. Des.*, vol. 2, no. 2, pp. 284–295, 2011, doi: 10.21825/scad.v2i2.20526.

[47] D. Steenbrugge and P. de Baets, "Aerodynamics of flue organ pipe voicing," *Int. J. Sustain. Construct. Des.*, vol. 1, no. 1, pp. 162–173, 2010, doi: 10.21825/scad.v1i1.20421.

[48] H.J. Außerlechner, T. Trommer, J. Angster, and A. Miklós, "Experimental jet velocity and edge tone investigations on a foot model of an organ pipe," *J. Acoust. Soc. Am.*, vol. 126, no. 2, pp. 878–886, Aug. 2009, doi: 10.1121/1.3158935.

[49] S.A. Elder, "On the mechanism of sound production in organ pipes," *J. Acoust. Soc. Am.*, vol. 54, no. 6, pp. 1554–1564, Dec. 1973, doi: 10.1121/1.1914453.

[50] D. Węgrzyn, P. Wrzeciono, and A. Wieczorkowska, "Recognition of the Flue Pipe Type Using Deep Learning," in *Intelligent Systems in Industrial Applications*. *ISMIS 2020*, M. Stettinger, G. Leitner, A. Felfernig and Z.W. Raś, Eds., Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-67148-8_7.

[51] J. Angster, S. Pitsch, and A. Miklós, "The influence of different types of windchests on the sound formation of flue organ pipes," *Fortschritte der Akustik*, vol. 30, pp. 1227–1228, 2004. [Online]. Available: https://pub.dega-akustik.de/DAGA_1999-2008/data/articles/001841.pdf. [Accessed: May 9, 2025].

[52] A.C. Lunn, "The influence of blowing pressure on pitch of organ pipes," *Phys. Rev.*, vol. 15, no. 5, pp. 446–449, May 1920, doi: 10.1103/PhysRev.15.446.

[53] B. Baretzky, M. Friesel, and B. Straumal, "Reconstruction of historical alloys for pipe organs brings true baroque music back to life," *MRS Bull.*, vol. 32, no. 3, pp. 249–255, Mar. 2007, doi: 10.1557/mrs2007.30.

[54] D. Steenbrugge, "Flue organ pipe operating regimes and voicing practices," in *Acoustics 2012*, Nantes, France: HAL, Apr. 2012, pp. 2789–2794. [Online]. Available: https://hal.science/hal-00811336. [Accessed: May 9, 2025].