World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:18, No:4, 2024

# Comparison of Machine Learning Techniques for Single Imputation on Audiograms

Sarah Beaver, Renee Bryce

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

*Abstract*—Audiograms detect hearing impairment, but missing values pose problems. This work explores imputations in an attempt to improve accuracy. This work implements Linear Regression, Lasso, Linear Support Vector Regression, Bayesian Ridge, K Nearest Neighbors (KNN), and Random Forest machine learning techniques to impute audiogram frequencies ranging from 125 Hz to 8000 Hz. The data contain patients who had or were candidates for cochlear implants. Accuracy is compared across two different Nested Cross-Validation k values. Over 4000 audiograms were used from 800 unique patients. Additionally, training on data combines and compares left and right ear audiograms versus single ear side audiograms. The accuracy achieved using Root Mean Square Error (RMSE) values for the best models for Random Forest ranges from 4.74 to 6.37. The $R^2$ values for the best models for Random Forest ranges from .91 to .96. The accuracy achieved using RMSE values for the best models for KNN ranges from 5.00 to 7.72. The $R^2$ values for the best models for KNN ranges from .89 to .95. The best imputation models received $R^2$ between .89 to .96 and RMSE values less than 8dB. We also show that the accuracy of classification predictive models performed better with our imputation models versus constant imputations by a two percent increase.

*Keywords*—Machine Learning, audiograms, data imputations, single imputations.

## I. Introduction

MISSING data in a dataset denotes the lack of information, presenting challenges in statistical analyses and interpretation. A common strategy to tackle this issue is employing single imputations, where a missing value is replaced with a single estimated value. In the context of audiograms, the absence of data poses a significant hurdle, especially in predictive modeling within audiology. Audiograms may have gaps due to factors like incomplete testing or patients' non-responsiveness to certain tones. These data gaps hinder the development and precision of predictive models aimed at anticipating hearing trends or evaluating intervention effectiveness. The deficiency of crucial information limits the model's capacity for reliable predictions, impeding progress in comprehending and addressing hearing impairments. Effectively addressing the missing data challenge in audiograms is paramount for improving the effectiveness of predictive modeling in audiology, a goal that can be achieved through the application of machine learning techniques for single imputations.

The National Health and Nutrition Examination Surveys (NHANES) conducted from 1999 to 2004 reported a prevalence rate of 16.1% of the United States national population with hearing loss using audio metric testing [1]. Audiograms play a critical role in clinical diagnosis and treatment of hearing loss [2]. Audiograms are the results of hearing test that shows different frequencies and intensity (decibels) levels that are usually portrayed in a graphical format [3]. These frequencies can range from 125Hz up to 8000Hz with decibel ranges from 0 dB to 120 dB [4].

Many reasons for missing data in clinical settings include but are not limited to human error with mistaken data entry, incomplete features, and patient refusal: which are typically identified by blanks, impossible values, nulls, and more [5], [6]. Additionally, audiograms can have missing values due to clinics having different protocols and prioritizing different frequencies. Audiograms with missing values are typically discarded or imputed using a basic mean of adjacent frequencies [2]. Machine learning techniques used for imputations are becoming more prevalent in clinical settings. In this article, we present different machine learning techniques that can be used for single imputation of audiograms at different frequencies.

The remainder of this paper is split into six more sections. Section II covers the background section, which reviews existing literature. Section III covers the research methodology, which includes discussions about the database and data, machine learning, and imputation models. Section IV discusses the results of the imputation models, Section V considers threats to the validity of this paper, and Section VI discusses the conclusion. Finally, Section VII discusses future works.

## II. Background

Statistical imputation techniques versus machine learning imputations on the breast cancer problem were compared in [7]. Researchers used univariate mean imputations, hot deck imputations, and multiple imputation software packages, including SAS and MICE, for the statistical imputation techniques, while the machine learning imputation techniques include multi-layer perceptron and k nearest neighbor (KNN) in [7]. They found that machine learning techniques were the most suited for imputing missing values, which led to a significant enhancement of prognosis accuracy compared to imputation methods based on statistical procedures [7].

Machine learning based imputation method KNN can be seen used in [8] where they use different KNN impute methods on clustering for missing gene values. A second machine learning based imputation method linear regression can be seen used in [9], where they worked on missing data from

Sarah Beaver and Renee Bryce are with Computer Science & Engineering, University of North Texas, USA (e-mail: Sarah.Beaver@unt.edu, Renee.Bryce@unt.edu).

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

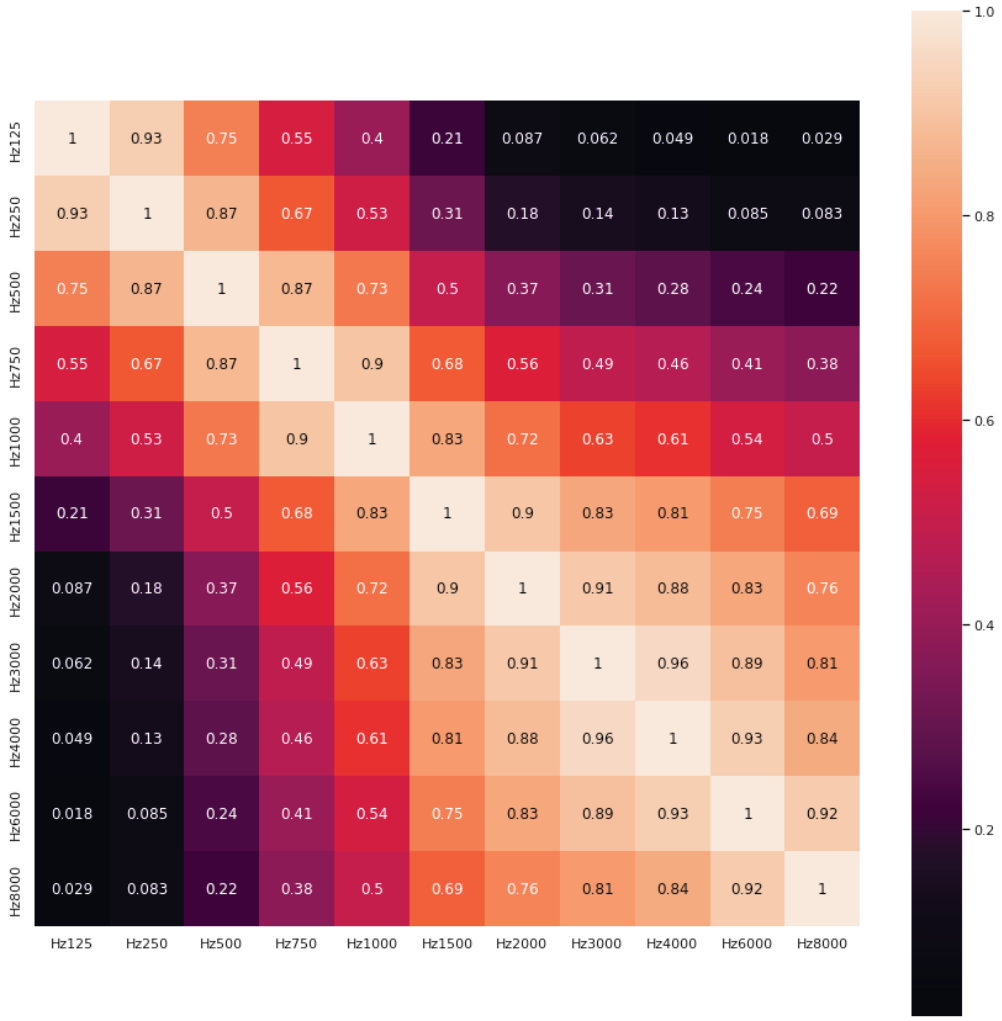


Fig. 1 Correlation between frequencies of Audiograms conducted by Spearman correlation

higher education institutions and looked at the accuracy of the imputation by comparing imputed and original values. The assumption was made that data should keep the same distribution.

Imputation methods have also been done on hearing data, as seen in [10], where they work on missing data from adolescents in a comprehensive leisure noise exposure study using multiple imputations. Future studies would include comparing multiple imputation techniques and expanding to include speech and word recognition tests. The use of a machine learning imputation technique on speech and word recognition tests can be seen in the next paragraph.

Imputation techniques have been performed on HERMES dataset previously, where linear regression was used to impute for the speech recognition test of CNCw and AZBio. These researchers had 430 cochlear implanted ears with an average absolute difference between observed and imputed CNCw was 10.5% in [11].

An example of machine learning techniques used with audiograms, where a Gaussian regression process is used to provide a real-time estimate of pure tone detection in humans. The real-time estimate can produce continuous audiograms accurately. Researchers used 21 patients with a mean absolute difference on the threshold of 5dBHL which can be seen in [12] and [13]. These works do not work for imputing missing data in existing audiograms.

A common approach to handling missing data in audiograms is to use the slope of completed audiograms similar to the audiogram with the missing value as seen in [14], [15]. The standard deviation levels greater than 15 decibels of hearing level (dBHL), considered a high standard deviation for audiograms [15], demonstrate the common approach for missing audiogram data is poor.

Imputations have been worked on in audiograms before, as seen with [16]. In this work, the authors compare KNN, Decision Tree, Random Forest, and multi-layer perceptron imputation for audiograms for children. The authors indicated three problems as follows. The most significant problem is regarding the small size of their data set. They only have 206 audiograms, which makes it hard to train machine learning algorithms. One reason they have such a small number of audiograms is that they focus on children under five with sensorineural hearing loss. The small number of audiograms leads to the second problem: their machine learning imputation

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

techniques are only good for children of this age group with sensorineural hearing loss. Lastly, the children's audiograms could have been completed at later visits to the clinic instead of during one session. Having an audiogram result from multiple sessions could cause variability from unknown events between sessions. The machine learning's average absolute threshold differences were up to 10 dBHL.

A KNN approach has been used to impute audiogram values for 3000 Hz and 6000 Hz. Researchers used a Gaussian Mixture Model (GMM) and a KNN model, which showed an improvement of imputation for 6000Hz over simple averaging method [2].

Imputations on audiogram data sets have also been studied which can be seen in [4]. These researchers have shown and discussed the importance of imputations in audiograms. The authors tested multiple imputation techniques on the audiograms, such as multiple imputations by chained equations (MICE), KNN, Neural Networks, univariate imputation, and gradient boosted trees.

Our work expands by increasing the number of machine learning imputation techniques we compare. Additionally, we have a few thousand audiograms, which is a much better number for training the machine learning models. Our audiograms come from adults with hearing loss, but the hearing loss type is not limited to sensorineural as in [16]. Since our data set contains adult patients, we do not have the issue of a lack of cooperation that these authors had with children; most adults can sit through an entire session quietly and calmly. Our work does single imputations for each frequency instead of taking them all simultaneously, as seen in [4]. Additionally, we compare the accuracy of classifier models where data were imputed with constant imputations versus imputing with our top two performing machine learning imputation models.

## III. Research Methodology

### A. Data and Pre-processing

The data were provided by HIPAA-secure, Encrypted, Research, Management, and Evaluation Solution (HERMES). It is a web-based Cochlear Implant (CI) database created by the nonprofit Auditory Implant Initiative (AII) [17]. Data include demographics (age, sex, ethnicity, geographic location), otologic history (etiology and duration of Hearing Loss (HL) preimplantation, chronic ear conditions, previous hearing aid use), relevant medical/surgical history, preoperative and implantation history (audiological results during candidacy evaluation, preoperative tinnitus, and vertigo, imaging, CI manufacturer and electrode type, operative technique, duration of electrode insertion, surgeon experience, intraoperative complications, antibiotics, and steroid use) and postoperative outcome data (postoperative audiological results and complications). The HERMES data set includes patients who have undergone cochlear implantation, CI candidates awaiting implantation, and patients with HL who may not yet be candidates [18].

The missing data can be summarized into three types [19], [5], [6]:

TABLE I
Dataset Summarization

| | |
|---|---|
| **Male** | **2530** |
| **Female** | **2082** |
| **Age of visit range** | **18 years old - 91 years old** |
| **Right** | **2337** |
| **Left** | **2281** |
| **Number of Unique Patients** | **803** |

1) Missing Completely At Random (MCAR): is when there is missing data randomly because the data do not exhibit an identifiable pattern. The more uniform the data distribution is, the less bias is expected to be introduced in the database.
2) Missing At Random (MAR): is when the missing data does have an identifiable pattern. MAR data means we can find a common factor with the missing data in the database.
3) Missing Not At Random (MNAR): is comparable to MAR, but the values causing others to be missing are not known.

For the HERMES data set, we can assume MAR because of the high correlation among frequencies of audiograms, which can be seen in Fig. 1. The reason for missing data can be explained by different clinics testing different frequencies, having different protocols for each doctor or clinic, and human error.

Data were selected if each of the ten frequencies for 125 HZ, 250 HZ, 500 HZ, 750 HZ, 1000 HZ, 2000 HZ, 3000 HZ, 4000 HZ, 6000 HZ, 8000 HZ did not have a missing value for training and testing the machine learning models, age at the visit was greater than 18 years, and if the conduction for the testing used was air. Conduction type was limited to air because there exist few audiograms with a conduction type of bone (n ¡ 100). The selection criteria gave 4618 audiograms to train and test machine learning imputation techniques. Other data included were the age at the visit, which ear was tested, hearing conditions of the ears (such as plugged, hearing aid, cochlear implant, and unaided), and gender. A table describing the data can be seen in Table I.

Outliers were removed based on box-plots interquartile range (IQR). This technique considers anything outside the range of Q1-1.5*(Q3-Q1) and Q3+1.5*(Q3-Q1) to be an outlier, where Q1 is the first quartile, and Q3 is the third quartile. The IQR outlier detection method is not as affected by extreme anomalies, as discussed in [20]. After removing values outside the IQR range, we were left with 3468 audiograms.

The correlation between frequencies among the rows can be seen in Fig. 1, conducted using Spearman on the remaining 3468 audiograms data after converting categorical data into individual columns. The Spearman correlation shows that frequencies close to each other have a high correlation, implying that these highly correlated features can be used to create machine learning techniques for imputing missing or invalid data points. An example is that the frequency 500 Hz

TABLE II
MODEL GRID SEARCH PARAMETERS

| Model | Parameter | Options |
|---|---|---|
| **Linear Regression** | Fitting Intercept | True, False |
| | Coefficient Forced Positive | True, False |
| **Ridge Regression** | Alpha | 0.5, 1, 2, 5, 10, 50 |
| | Fitting Intercept | True, False |
| | Reuse Previous Run Solution | True, False |
| | Max Iterations | 1000, 1500, 2000, 2500, 5000 |
| | Coefficient Selection | cyclic, random |
| | Precomputed Gram Matrix | True, False |
| **LASSO** | Alpha | 0.5, 1, 2, 5, 10, 50 |
| | Fitting Intercept | True, False |
| | Coefficient Selection | cyclic, random |
| | Precomputed Gram Matrix | True, False |
| | Max Iterations | 1000, 1500, 2000, 2500, 5000 |
| | Reuse Previous Run Solution | True, False |
| **Bayesian Ridge** | First Alpha | 1e-6, 5e-6, 1e-5, 5e-5, 1e-7, 5e-7 |
| | Second Alpha | 1e-6, 5e-6, 1e-5, 5e-5, 1e-7, 5e-7 |
| | Fitting Intercept | True, False |
| | First Lambda | 1e-6, 5e-6, 1e-5, 5e-5, 1e-7, 5e-7 |
| | Second Lambda | 1e-6, 5e-6, 1e-5, 5e-5, 1e-7, 5e-7 |
| | Number of Iterations | 200, 250, 300, 400, 500 |
| **Linear Support Vector Regression** | Tolerance | 1e-3, 1e-4, 1e-5 |
| | Fitting Intercept | True, False |
| | C Parameter | 0.001, 0.01, 0.1, 1, 10, 100, 1000 |
| | Loss Function | L1 |
| | Dual Optimization | False since number of samples greater than number of features |
| **K Nearest Neighbors** | Weight | uniform, distance based on list (auto, ball tree, KD tree, brute) |
| | Leaf Size for Ball Tree | 1 to 50 (Increment of 1) |
| | Distance | Manhattan, Euclidean |
| | Number of Neighbors | 1 to 50 (Increment of 1) |
| **Random Forest Regression** | Number of Estimators | 50, 100, 200, 250 |
| | Error Criterion | Squared, Absolute, Poisson |
| | Max Depth | None, 3, 9, 12 |
| | Max feature | Auto, Sq. root, Log 2, 2, 3, None |
| | Bootstrap | Yes, No |

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

has a correlation of 0.86 to the frequency 250 Hz and 0.87 to the frequency 750 Hz, while it has a correlation of 0.35 to the frequency 2000 Hz. Upper frequencies have a lower correlation to adjacent frequencies, which can be seen with frequency 6000 Hz's correlation of 0.36 to frequency 8000 Hz.

### *B. Machine Learning*

Python version 3.8 was used as the programming language, and Scikit-learn library [21] was used for models. The models created were Linear regression, Ridge Regression, Lasso, Bayesian Ridge, Linear Support Vector Regression, KNN, and Random Forest regression.

*Linear regression*: A statistical technique known as linear regression can be implemented to simulate the linear connection between a dependent variable and at least one independent variable. The premise of linear regression is that there is a straight line that can be drawn to represent the connection between the dependent and independent variable(s). Finding the best-fit line that minimizes the difference between the predicted values and the actual values of the dependent variable is the objective of linear regression.

*Ridge regression*: Ridge regression is a type of linear regression used when the data suffer from multicollinearity, which occurs when the independent variables are highly correlated. In ridge regression, a penalty term is added to the

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

sum of squared residuals, which helps to prevent overfitting by reducing the magnitude of the regression coefficients. This penalty term is controlled by a hyperparameter called the regularization parameter or lambda, which determines the amount of shrinkage applied to the coefficients.

*LASSO*: LASSO, which stands for Least Absolute Shrinkage and Selection Operator, is a type of linear regression that adds a penalty to the cost function to shrink the input variables' coefficients toward zero. This penalty encourages sparsity in the solution, meaning it can be used for feature selection and prediction.

*Bayesian Ridge*: Bayesian Ridge is a linear regression model that uses Bayesian methods for regularization. It is a regularized linear regression version that can handle multicollinearity and overfitting. In Bayesian Ridge, a prior distribution is placed on the model coefficients, which helps to regularize the model and prevent overfitting. The prior distribution is assumed to be Gaussian, with zero mean and a precision parameter. The precision parameter is a hyperparameter that controls the strength of the regularization.

*Linear Support Vector Regression*: Linear Support Vector Regression (SVR) is a regression algorithm that uses Support Vector Machines (SVMs) to perform regression analysis. Like other SVM-based algorithms, SVR finds a linear function that best fits the data by maximizing the margin between predicted and actual values. The margin is the distance between the nearest data points and the hyperplane.

*KNN*: KNN is a non-parametric regression algorithm for classification and regression tasks. In KNN, the output is a class membership or a real value prediction based on the k-nearest examples in the training data set. The algorithm calculates the distance between the new data point and all the training examples. Then, the k-nearest training examples are selected based on the shortest distance to the new data point. Finally, the output value is computed by averaging the values of the k-nearest neighbors for regression or by a majority vote for classification. The value of k is the number of neighbors to be considered, a hyperparameter that must be set beforehand.

*Random Forest Regression*: Random Forest regression is a popular ensemble learning method for regression tasks. It involves creating multiple decision trees and aggregating their predictions to make a final prediction. Each tree is built on a random subset of the data and features, which helps to reduce overfitting. Each tree produces a prediction during prediction, and the final prediction is the average of all the predictions.

### C. Imputation Models

Imputation models were created for each frequency for each ML technique tested on the audiogram. Nested Cross-Validation was used, as discussed in [22]. Hyperparameter optimization will pick the model that has the best accuracy, which can cause overfitting. Nested Cross-Validation helps find a good trade-off between bias and variance to prevent overfitting, as discussed in [22]. An example of a possible pseudo-code for nested cross-validation is shown in algorithm 1.

Two variations of nested cross-validation were used. The first was with the outer cross-validation using a split of 10, and the inner cross-validation used a split of 5. The second is the reverse, with the outer cross-validation using a split of 5 and the inner using a split of 10. Models were created using both ears together, left ear only or right ear only, which gives a total of six models per frequency per ML technique. A grid search was performed on each model to find the best parameters per model per frequency, as shown in Table II.

**Algorithm 1** Nested Cross-Validation Pseudo Code

**Require:** Dataset $D$
**Require:** Number of outer folds $k_{out}$
**Require:** Number of inner folds $k_{in}$
**Require:** Model $M$
**Require:** Evaluation metric $E$
Divide $D$ into $k_{out}$ outer folds of $D_1, D_2, \ldots, D_{k_{out}}$
**for** $i \leftarrow 1$ to $k_{out}$ **do**
    Set aside $D_i$ as the test set
    Combine the remaining outer folds into the training set $T$
    Divide $T$ into $k_{in}$ inner folds of $T_1, T_2, \ldots, T_{k_{in}}$
    **for** $j \leftarrow 1$ to $k_{in}$ **do**
        Set aside $T_j$ as the validation set
        Combine the remaining inner folds into the training set $t$
        Train the model $M$ on $t$
        Evaluate $M$ on $T_j$ using evaluation metric $E$ and record the performance
    **end for**
    Choose the best hyperparameters for $M$ based on the inner cross-validation
    Train $M$ on the combined inner folds $T$
    Evaluate $M$ on the test set $D_i$ using evaluation metric $E$ and record the performance
**end for**
Compute the overall performance of $M$ based on the $k_{out}$ test folds

### D. Performance Evaluation

We compare the regression of the models in two parts. The first is $R^2$, (also known as R-squared or coefficient of determination), hereafter referred to as $R^2$, which gives a measure of variance between dependent and independent variables. The mathematical properties of why $R^2$ is a standard metric for regression models and explanations of the interpretability can be seen in [23]. The equation for $R^2$ can be seen in (1):

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{m}(Y_i - \bar{Y}_i)^2} \quad (1)$$

where $Y_i$ = the actual value; $\hat{Y}_i$ = the predicted value; $\bar{Y}_i$ = mean of the actual values.

The second measure was using the root-mean-squared error (RMSE) score. The good thing about RMSE is that we can see the metrics in the same unit as our prediction; in the case of our works, that would be decibels. Many papers, including [24] and [25], discussed cases where RMSE is a useful metric

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

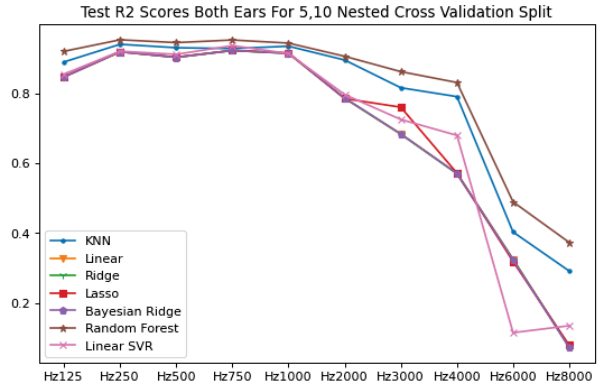


Fig. 2 Test $R^2$ Scores For Both Ears with nested cross-validation of 10,5

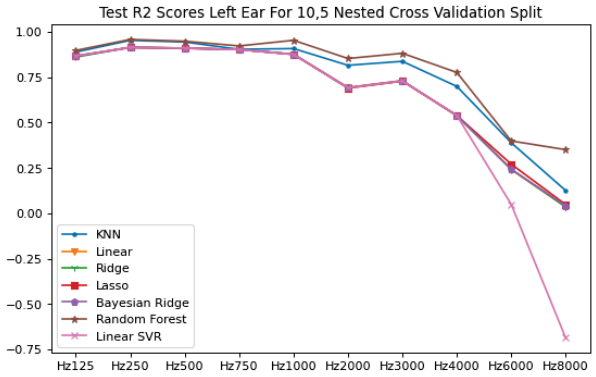


Fig. 3 Test $R^2$ Scores For Left Ear with nested cross-validation of 10,5

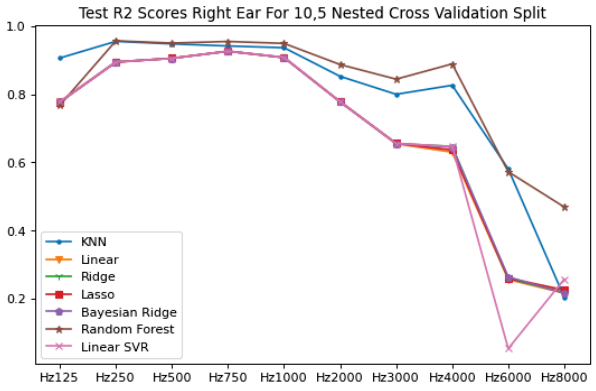


Fig. 4 Test $R^2$ Scores For Right Ear with nested cross-validation of 10,5

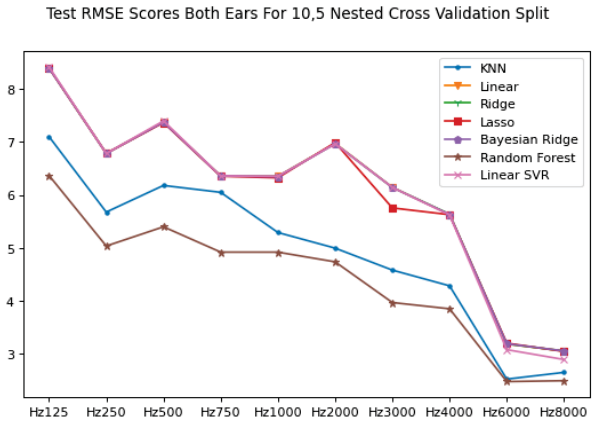


Fig. 5 Test RMSE Scores For Both Ears with nested cross-validation of 10,5

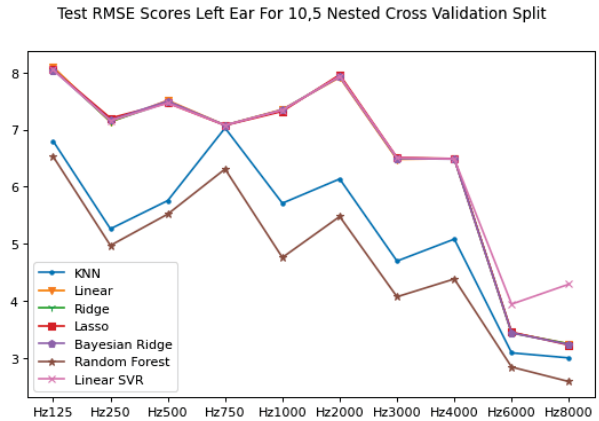


Fig. 6 Test RMSE Scores For Left Ear with nested cross-validation of 10,5

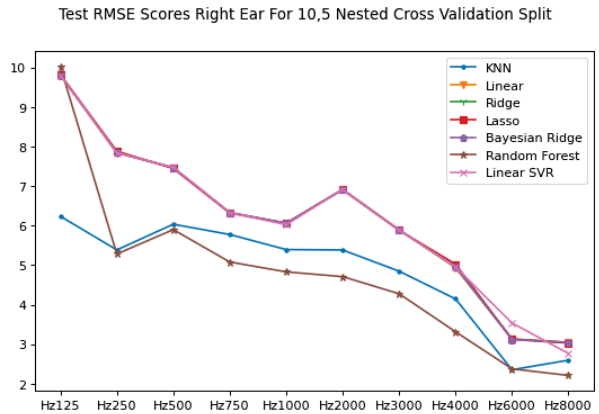


Fig. 7 Test RMSE Scores For Right Ear with nested cross-validation of 10,5

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

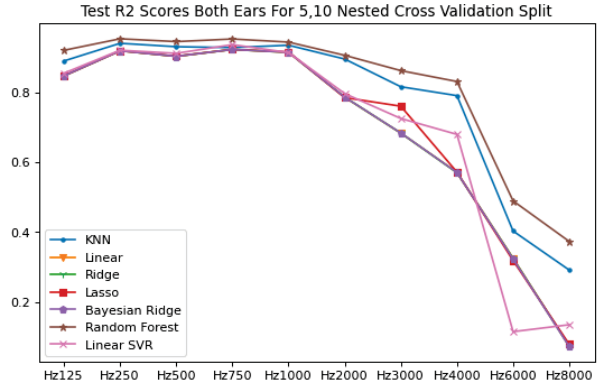


Fig. 8 Test $R^2$ Scores For Both Ears with nested cross-validation of 5,10

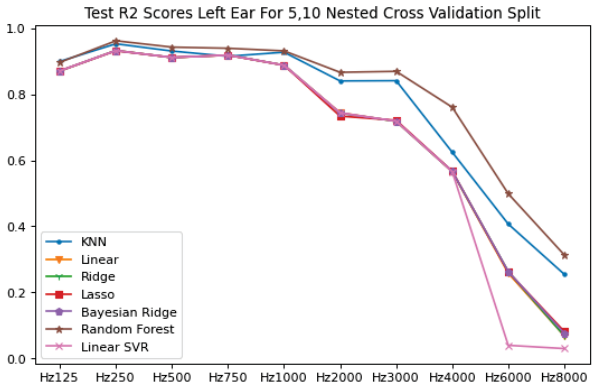


Fig. 9 Test $R^2$ Scores For Left Ear with nested cross-validation of 5,10

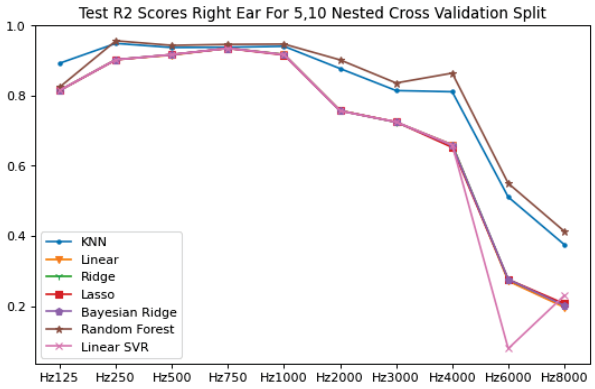


Fig. 10 Test $R^2$ Scores For Right Ear with nested cross-validation of 5,10

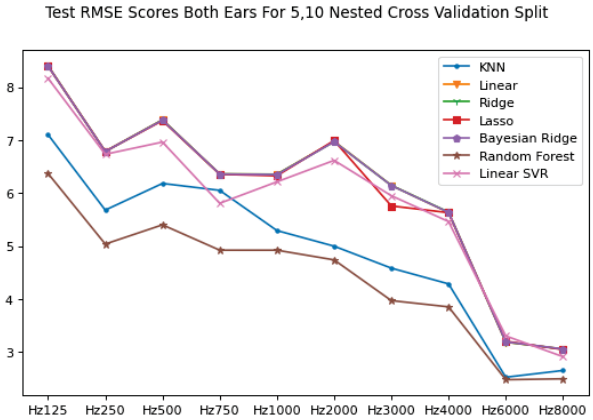


Fig. 11 Test RMSE Scores For Both Ears with nested cross-validation of 5,10

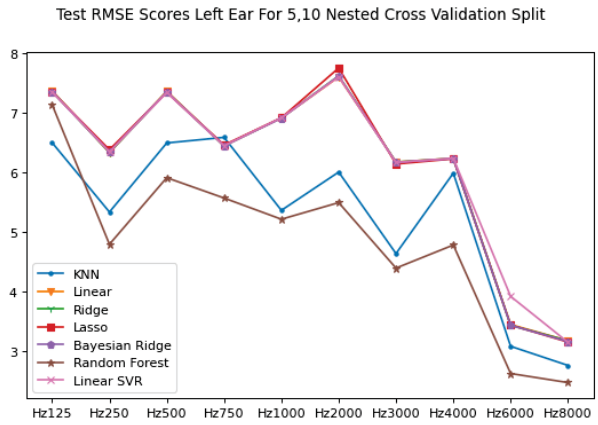


Fig. 12 Test RMSE Scores For Left Ear with nested cross-validation of 5,10

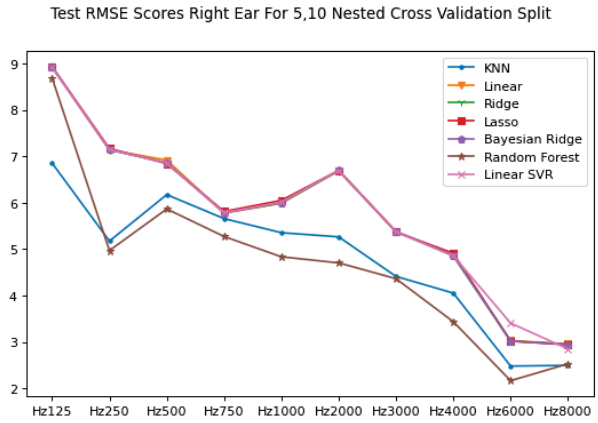


Fig. 13 Test RMSE Scores For Right Ear with nested cross-validation of 5,10

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

but state it should be used in combination with other metrics, which we are doing here with $R^2$. The equation to RMSE can be seen in (2):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

where $Y_i$ = the actual value; $\hat{Y}_i$ = the predicted value.

## IV. Results

The results are split into four sections, with the first two sections based on the two different nested cross-validation parameters and based on comparing RMSE and $R^2$ scores. The first section contains results with nested cross-validation as 10 for the outer loop and 5 for the inner loop for $R^2$ scores followed by RMSE scores. The second set has nested cross-validation results of 5 for the outer loop and 10 for the inner loop for $R^2$, followed by RMSE scores. The third section shows the results of the top two imputation models. The fourth section compares the top two model imputations to a constant imputation technique for accuracy on simple prediction models.

The first set of results is shown with the cross-validation having an outer loop of 10 and an inner loop of 5. There are three results for using both ears, the left ear only, or the right ear only, based on RMSE scores, and three for both ears, the left ear only, or the right ear only, based on $R^2$ score. The RMSE results show the score for each frequency, which can be seen in Figs. 5, 6, 7 and the $R^2$ results can be seen in Figs. 2, 3, 4 for each frequency.

The second set of results is shown with the cross-validation having an outer loop of 5 and an inner loop of 10. There are three results for using both ears, the left ear only, or the right ear only, based on RMSE scores, and three for both ears, left ear only, or right ear only, based on $R^2$ score. The RMSE results show the score for each frequency, which can be seen in Figs. 11, 12, 13 and the $R^2$ results can be seen in Figs. 8, 9, 10 for each frequency.

The third section shows that the graphs show KNN and Random Forest have the highest $R^2$ and lowest RMSE scores across frequencies. As the frequencies increase, the $R^2$ scores decrease, but the RMSE scores increase. The smaller ranges can explain the scores discrepancies in values and less variance for upper frequencies starting with 4000 Hz. The range of 4000 Hz is 80 decibels to 125 decibels, the range of 6000 Hz is 95 decibels to 110 decibels and the range of 8000 Hz is 85 decibels to 100 decibels. Since the ranges are getting smaller, it means there is less variance in the upper frequencies, thus giving us higher RMSE scores. Even though there are high RMSE scores in upper frequencies, there are also very low $R^2$ scores, which means that the model cannot predict the outcome well for that frequency. Therefore, the models for the upper frequencies are not fitting well, while the rest are performing effectively.

We can see that KNN and Random Forest perform the best across audiogram frequencies and cross-validation combinations. A closer look at the results of Random Forest models can be seen in Tables III and IV. The results of the KNN models can be seen in Tables V and VI. The best for each frequency for each technique has been bolded. The best models were decided by selecting the model with the highest $R^2$ score and the lowest RMSE score, even though the results are mostly similar across each result. Since decibels increase in increments of 5, seeing RMSE scores averaging 5 points shows that the models can predict within one shift of the actual value. These make the models great for imputation in these frequencies using models with high $R^2$ score. In KNN, the $R^2$ score drops by less than or equal to 0.05 in frequencies 125 Hz to 3000 Hz in comparable models, and RMSE scores have a variance of 1.5 from the best model. Random Forest's $R^2$ score also shows a drop less than or equal to 0.05 in frequencies 125 Hz to 3000 Hz, and RMSE scores show a variance range of 1.5 from the best model. Random Forest performs slightly better when compared to KNN but is still comparable. Depending on time, KNN could be used because of its faster training time if some sensitivity on data imputation can be given up.

The last section discusses how we compared Random Forest and KNN imputation models on improved accuracy against constant imputation. We used the hyperparameters found for both ears from nested cross-validation of an outer loop of 5 and an inner loop of 10. The models were fitted on the dataset without missing data. For the constant imputation, null values were imputed with negative ones. After that, we group audio conditions to form a binary label that the predictive models predicted. The label was created with the first group containing patients if their hearing condition was plugged for the left ear and unaided for the right ear, or unaided for the left ear and plugged for the right ear. The second group contained patients whose hearing condition has a Cochlear Implant for the left ear and plugged for the right ear, or plugged for the right ear and has a Cochlear Implant for the left ear. In other words, the label was based on whether a patient's hearing condition had a cochlear implant in the tested ear and plugged in the untested ear during the pure-tone testing for the audiogram. The data set was expanded to 6825 rows after imputing rows where one frequency of the ten was missing and if the hearing conditions only included plugged, unaided, or Cochlear Implant combinations to build the groups. Other data include audiogram frequencies, gender, hearing condition, and age at the visit.

The two models that were built were a KNN Classifier and a Logistic Regression with a hyperparameter of max iteration of 1500. Cross-validation was performed with stratified k = 10 to keep an equal portion when training. We computed the mean score and standard deviation for the predictive models after running each model 50 times. These results can be seen in Table VII. For the KNN Classifier, there is a two percent increase in accuracy greater than 1.5 the standard deviation meaning this small increase is indeed significant. Further model tuning could lead to an even bigger accuracy difference between the imputations types. The improvement in accuracy shows that proper imputation models can be used to improve other predictive models' accuracy while increasing dataset size.

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

TABLE III
RANDOM FOREST IMPUTATIONS RMSE AND $R^2$ SCORES ACROSS FREQUENCIES FOR 5,10 NESTED CROSS-VALIDATION

| | **Random Forest 5,10 cross-validation** | | | | | |
|---|---|---|---|---|---|---|
| | **Both Ears 5,10 cross-validation** | | **Left Ear 5,10 cross-validation** | | **Right Ear 5,10 cross-validation** | |
| | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** |
| **Hz125** | **6.27** | **0.91** | 7.14 | 0.90 | 8.69 | 0.82 |
| **Hz250** | 4.90 | 0.96 | **4.79** | **0.96** | 4.97 | 0.96 |
| **Hz500** | **5.70** | **0.94** | 5.91 | 0.94 | 5.87 | 0.94 |
| **Hz750** | **4.88** | **0.95** | 5.57 | 0.94 | 5.27 | 0.95 |
| **Hz1000** | 4.88 | 0.95 | 5.21 | 0.93 | **4.84** | **0.95** |
| **Hz2000** | 4.89 | 0.89 | 5.49 | 0.87 | **4.7** | **0.90** |
| **Hz3000** | 4.16 | 0.85 | **4.39** | **0.87** | 4.36 | 0.84 |
| **Hz4000** | 4.02 | 0.81 | 4.78 | 0.76 | **3.44** | **0.86** |
| **Hz6000** | 2.43 | 0.52 | 2.62 | 0.50 | **2.17** | **0.55** |
| **Hz8000** | 2.47 | 0.39 | 2.47 | 0.31 | **2.53** | **0.41** |

TABLE IV
RANDOM FOREST IMPUTATIONS RMSE AND $R^2$ SCORES ACROSS FREQUENCIES FOR 10,5 NESTED CROSS-VALIDATION

| | **Random Forest 10,5 cross-validation** | | | | | |
|---|---|---|---|---|---|---|
| | **Both Ears 10,5 cross-validation** | | **Left Ear 10,5 cross-validation** | | **Right Ear 10,5 cross-validation** | |
| | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** |
| **Hz125** | **6.37** | **.92** | 6.53 | 0.90 | 10.02 | 0.77 |
| **Hz250** | 5.04 | 0.95 | **4.97** | **0.96** | 5.28 | 0.96 |
| **Hz500** | 5.40 | 0.94 | 5.52 | 0.90 | **5.90** | **0.95** |
| **Hz750** | **4.92** | **0.95** | 6.31 | 0.92 | 5.08 | 0.95 |
| **Hz1000** | 4.92 | 0.94 | **4.76** | **0.95** | 4.83 | 0.95 |
| **Hz2000** | **4.74** | **0.91** | 5.48 | 0.85 | 4.71 | 0.89 |
| **Hz3000** | 3.97 | 0.86 | **4.07** | **0.88** | 4.28 | 0.84 |
| **Hz4000** | 3.85 | 0.83 | 4.38 | 0.78 | **3.31** | **0.89** |
| **Hz6000** | 2.47 | 0.49 | 2.84 | 0.40 | **2.38** | **0.57** |
| **Hz8000** | 2.50 | 0.37 | 2.58 | 0.35 | **2.21** | **0.47** |

TABLE V
KNN IMPUTATIONS RMSE AND $R^2$ SCORES ACROSS FREQUENCIES FOR 5,10 NESTED CROSS-VALIDATION

| | **KNN 5,10 cross-validation** | | | | | |
|---|---|---|---|---|---|---|
| | **Both Ears 5,10 cross-validation** | | **Left Ear 5,10 cross-validation** | | **Right Ear 5,10 cross-validation** | |
| | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** | **RMSE** | **$R^2$** |
| **Hz125** | 7.72 | 0.87 | **6.50** | **0.90** | 6.85 | 0.89 |
| **Hz250** | 5.57 | 0.94 | 5.33 | 0.95 | **5.17** | **0.95** |
| **Hz500** | 6.33 | 0.93 | 6.49 | 0.93 | **6.17** | **0.94** |
| **Hz750** | 5.70 | 0.94 | 6.59 | 0.92 | **5.66** | **0.94** |
| **Hz1000** | **5.00** | **0.94** | 5.36 | 0.93 | 5.35 | 0.94 |
| **Hz2000** | **5.03** | **0.88** | 6.00 | 0.84 | 5.26 | 0.88 |
| **Hz3000** | 4.70 | 0.81 | **4.63** | **0.84** | 5.38 | 0.72 |
| **Hz4000** | 4.63 | 0.75 | 5.99 | 0.62 | **4.05** | **0.81** |
| **Hz6000** | 2.64 | 0.43 | 3.08 | 0.41 | **2.48** | **0.51** |
| **Hz8000** | 2.67 | 0.29 | 2.76 | 0.25 | **2.50** | **0.38** |

TABLE VI
KNN IMPUTATIONS RMSE AND $R^2$ SCORES ACROSS FREQUENCIES FOR 10,5 NESTED CROSS-VALIDATION

| | KNN 10,5 cross-validation | | | | | |
|---|---|---|---|---|---|---|
| | Both Ears 10,5 cross-validation | | Left Ear 10,5 cross-validation | | Right Ear 10,5 cross-validation | |
| | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ |
| **Hz125** | 7.10 | 0.89 | 6.80 | 0.89 | **6.24** | **0.91** |
| **Hz250** | 5.68 | 0.94 | **5.26** | **0.95** | 5.38 | 0.95 |
| **Hz500** | 6.18 | 0.93 | 5.75 | 0.94 | **6.03** | **0.95** |
| **Hz750** | 6.05 | 0.93 | 7.03 | 0.90 | **5.77** | **0.94** |
| **Hz1000** | 5.29 | 0.93 | 5.71 | 0.91 | **5.39** | **0.94** |
| **Hz2000** | **5.00** | **0.89** | 6.13 | 0.82 | 5.38 | 0.85 |
| **Hz3000** | 4.58 | 0.82 | **4.7** | **0.84** | 4.85 | 0.80 |
| **Hz4000** | 4.28 | 0.79 | 5.08 | 0.70 | **4.14** | **0.83** |
| **Hz6000** | 2.52 | 0.40 | 3.09 | 0.39 | **2.35** | **0.58** |
| **Hz8000** | **2.65** | **0.29** | 3.00 | 0.13 | 2.60 | 0.20 |

TABLE VII
PREDICTIVE MODEL ACCURACY SCORES AFTER IMPUTATIONS

| Imputation Technique | KNN Classifier | | Logistic Regression | |
|---|---|---|---|---|
| | Mean Score | Standard Deviation | Mean Score | Standard Deviation |
| **Constant Imputation** | 86.40% | 0.010 | 79.36% | 0.0071 |
| **Random Forest Imputation** | 88.40% | 0.009 | 80.25% | 0.0067 |
| **KNN Imputation** | 88.52% | 0.009 | 80.20% | 0.0066 |

## V. THREATS TO VALIDITY

Different data sets could lead to different results or have a different machine learning technique perform better. Having more data or data variety, such as data containing bone conduction or hearing aids for ear configuration, could increase the validity of the research. Cross-validation was used to help prevent overfitting in the models, especially with Random Forest. Two types of cross-validation were tested. Models were run multiple times, getting similar results each time (for some runs, KNN would perform closer to or slightly better than Random Forest for a frequency). This paper shows one set of those results.

## VI. CONCLUSIONS

This work applies machine learning models to impute missing frequency values of audiograms, which can impute the missing frequencies with RMSE values of less the 7dB for the best models while still having $R^2$ greater than .90. Imputed values have the potential to be used in machine learning models to predict Unaided or Cochlear Implants for patients. The imputation models in this work can be used as a baseline to implement more complicated techniques for better precision of imputation on audiograms and other medical data.

Our study's best single imputation methods, KNN and Random Forest, had RMSE values lower than seven compared to [4] whose best multiple imputation method using MICE that had RMSE values greater than 7. When comparing $R^2$, this study has comparable frequencies ranging from 125 Hz to 1000 Hz.

## VII. FUTURE WORK

Future work may examine a larger set of audiograms or other medical data sets with missing data. For instance, the imputation techniques may help with other missing audiometric data, including AzBio, CNC, and BKB. There is an opportunity to broaden the examination of parameters used in this work, especially with Random Forest. Different algorithms are another direction where improvements such as boosting techniques may be possible. This work will be used as a basis for creating a new data imputation technique which will be tested on the HERMES dataset.

## REFERENCES

[1] H. Mahboubi, H. W. Lin, and N. Bhattacharyya, "Prevalence, characteristics, and treatment patterns of hearing difficulty in the united states," *JAMA Otolaryngology–Head & Neck Surgery*, vol. 144, no. 1, pp. 65–70, 2018.

[2] F. Charih, A. Steeves, M. Bromwich, A. E. Mark, R. Lefrançois, and J. R. Green, "Applications of machine learning methods in retrospective studies on hearing," in *2018 IEEE Life Sciences Conference (LSC)*. IEEE, 2018, pp. 126–129.

[3] E. Rose, "Audiology," *Australian Family Physician*, vol. 40, no. 5, pp. 290–292, 2011.

[4] C. Pavelchek, A. P. Michelson, A. Walia, A. Ortmann, J. Herzog, C. A. Buchman, and M. A. Shew, "Imputation of missing values for cochlear implant candidate audiometric data and potential applications," *Plos one*, vol. 18, no. 2, p. e0281337, 2023.

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf

[5] M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das, "Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021)," *Informatics in Medicine Unlocked*, vol. 27, p. 100799, 2021.
[6] J. G. Ibrahim, H. Chu, and M.-H. Chen, "Missing data in clinical studies: issues and methods," *Journal of clinical oncology*, vol. 30, no. 26, p. 3297, 2012.
[7] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial intelligence in medicine*, vol. 50, no. 2, pp. 105–115, 2010.
[8] A. Dubey and A. Rasool, "Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.
[9] R. Bruni, C. Daraio, and D. Aureli, "Imputation techniques for the reconstruction of missing interconnected data from higher educational institutions," *Knowledge-Based Systems*, vol. 212, p. 106512, 2021.
[10] J. Wendl, D. Gerstner, J. Huß, V. Weilnhammer, C. Jenkac, C. Pérez-Àlvarez, T. Steffens, C. Herr, and S. Heinze, "Compensating for missing data in the ohrkan cohort study examining total leisure noise exposure among adolescents," *International Journal of Audiology*, vol. 61, no. 7, pp. 574–582, 2022.
[11] R. K. Sharma, S. Y. Chen, J. Grisel, and J. S. Golub, "Assessing cochlear implant performance in older adults using a single, universal outcome measure created with imputation in hermes," *Otology & Neurotology*, vol. 39, no. 8, pp. 987–994, 2018.
[12] D. L. Barbour, R. T. Howard, X. D. Song, N. Metzger, K. A. Sukesan, J. C. DiLorenzo, B. R. Snyder, J. Y. Chen, E. A. Degen, J. M. Buchbinder *et al.*, "Online machine learning audiometry," *Ear and hearing*, vol. 40, no. 4, p. 918, 2019.
[13] X. D. Song, B. M. Wallace, J. R. Gardner, N. M. Ledbetter, K. Q. Weinberger, and D. L. Barbour, "Fast, continuous audiogram estimation using machine learning," *Ear and hearing*, vol. 36, no. 6, p. e326, 2015.
[14] R. Feirn, "Guidelines for fitting hearing aids to young infants version 2.0 february 2014," 2014.
[15] A. L. Pittman and P. G. Stelmachowicz, "Hearing loss in children and adults: audiometric configuration, asymmetry, and progression," *Ear and hearing*, vol. 24, no. 3, p. 198, 2003.
[16] P. Pitathawatchai, S. Chaichulee, and V. Kirtsreesakul, "Robust machine learning method for imputing missing values in audiograms collected in children," *International Journal of Audiology*, vol. 61, no. 1, pp. 66–77, 2022.
[17] E. C. Schafer, J. J. Grisel, A. de Jong, K. Ravelo, A. Lam, M. Burke, T. Griffin, M. Winter, and D. Schrader, "Creating a framework for data sharing in cochlear implant research," *Cochlear Implants International*, vol. 17, no. 6, pp. 283–292, 2016.
[18] S. Y. Chen, J. J. Grisel, A. Lam, and J. S. Golub, "Assessing cochlear implant outcomes in older adults using hermes: A national web-based database," *Otology & Neurotology*, vol. 38, no. 10, pp. e405–e412, 2017.
[19] U. Garciarena and R. Santana, "An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers," *Expert Systems with Applications*, vol. 89, pp. 52–65, 2017.
[20] S. Saleem, M. Aslam, and M. R. Shaukat, "A review and empirical comparison of univariate outlier detection methods." *Pakistan Journal of Statistics*, vol. 37, no. 4, 2021.
[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[22] F. GUSTAFSSON, "Comparing random forest, xgboost and neural networks with hyperparameter optimization by nested cross-validation," 2019.
[23] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021.
[24] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.
[25] T. O. Hodson, "Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, 2022.

Open Science Index, Computer and Information Engineering Vol:18, No:4, 2024 publications.waset.org/10013606.pdf