Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 32, Num. 2 (2025) 52-63

RESEARCH ARTICLE

On the Use of Spectral Data from Smartphone Accelerometer Signals and Constituent Material for the Identification of Damaged Walls

Uso de Dados Espectrais de Sinais de Acelerômetro de Smartphone e Material Constituinte para a Identificação de Paredes Danificadas

Tales Boratto[1]*, Douglas Lima Fonseca[1], Heder Soares Bernardino[2]. Alex Borges[2], Alexandre Cury[3], Leonardo Goliatt[3]

Abstract: Monitoring the integrity of civil structures is crucial for ensuring their safety and longevity. Assessing the structural condition of walls is particularly important due to their role in stability and load distribution in buildings. Smartphones can be used to collect dynamic data from the walls of modern buildings. This strategy is an easier and cheaper way to obtain information concerning the wall's structural condition compared to other costly instrumentation plans using deflectomers, accelerometers, etc. Such a type of data was explored in the literature with good results. However, despite the quality of the models obtained, other features can be included to improve the results. For instance, spectral information may characterize the frequency content of a signal. Moreover, the material used to build the structure affects the signals collected. Thus, we propose the use of three machine learning models (Decision Tree, Random Forest, and K-Nearest Neighbors (KNN)) to identify damage in walls from vibration signals using their spectral data and the wall's constituent material in addition to those already used in the literature. The proposed improvements increased accuracy by about 23%, leading to an average accuracy of 97.78% with KNN when combining statistical and spectral features.
Keywords: Damage Detection — Structural Health Monitoring — Vibration Monitoring — Machine Learning

Resumo: O monitoramento da integridade das estruturas civis é fundamental para garantir sua segurança e longevidade. A avaliação da condição estrutural das paredes é particularmente importante devido à sua função na estabilidade e na distribuição de cargas nos edifícios. Os smartphones podem ser usados para coletar dados dinâmicos das paredes de edifícios modernos. Essa estratégia é uma maneira mais fácil e barata de obter informações sobre a condição estrutural da parede em comparação com outros planos de instrumentação caros que usam defletômetros, acelerômetros etc. Esse tipo de dado foi explorado na literatura com bons resultados. Entretanto, apesar da qualidade dos modelos obtidos, outros recursos podem ser incluídos para melhorar os resultados. Por exemplo, informações espectrais podem caracterizar o conteúdo de frequência de um sinal. Além disso, o material usado para construir a estrutura afeta os sinais coletados. Assim, propomos o uso de três modelos de aprendizado de máquina (Decision Tree, Random Forest e K-Nearest Neighbors (KNN)) para identificar danos em paredes a partir de sinais de vibração usando seus dados espectrais e o material constituinte da parede, além daqueles já usados na literatura. Os aprimoramentos propostos aumentaram a precisão em cerca de 23%, levando a uma precisão média de $97,78\%$ com o KNN ao combinar recursos estatísticos e espectrais.
Palavras-Chave: Detecção de Dano — Monitoramento de Integridade Estrutural — Monitoramento de Vibração — Aprendizado de Máquina

[1] *Graduate Program in Computational Modeling, Federal University of Juiz de Fora (UFJF), Brazil*
[2] *Department of Computer Science, Federal University of Juiz de Fora (UFJF), Brazil*
[3] *Department of Computational and Applied Mechanics, Federal University of Juiz de Fora (UFJF), Brazil*
***Corresponding author**: tales.boratto@estudante.ufjf.br
DOI: http://dx.doi.org/10.22456/2175-2745.142129 • **Received:** 26/08/2024 • **Accepted:** 16/01/2025

1. Introduction

Monitoring the structural condition of civil constructions is essential to ensuring their safety, reliability, and longevity. Such a strategy can identify possible early failures, reduce maintenance costs, and minimize the risk of collapse and catastrophic events [1]. In particular, monitoring the integrity of the walls of modern constructions, such as houses, buildings, and others, has also become a very relevant aspect since they play a fundamental role in distributing loads and maintaining structural stability. Therefore, certain types of damage, whether caused by overload, seismic events, thermal variation, water infiltration, etc., could compromise their functionality and cause an accident [2, 3].

In general, Structural Health Monitoring (SHM) techniques focus on two main approaches: vibration-based monitoring [4, 5], whose premise consists of the assumption that physical changes cause a change in the vibration characteristics of the structure, and vision-based monitoring, which is related to both visual inspection by experts and computer-automated inspection from images or videos [6, 7]. In the context of the first approach, previous studies have demonstrated their efficiency in identifying anomalies in different structures, such as bridges [8, 9], historic monument buildings [10, 11], among others [12, 13].

In parallel, considering that technological advances have enabled smartphones to be equipped with various types of sensors, including the well-known accelerometers, such devices have begun to be used as an alternative sensing instrumentation in vibration-based SHM investigations. Feng et al. [14] examined the viability of monitoring structural vibration under typical and high loads using smartphone accelerometers. Many shaking table experiments were carried out to compare these devices' sensors with high-quality accelerometers for sensing vibration at various frequencies. Finally, the authors also investigated the performance of smartphone sensors on a real pre-stressed reinforced concrete pedestrian bridge at Princeton (USA). Kang, Baek, and Park [15] used a smartphone application (app) *i-Jishin* to estimate the large deformation of a two-story building with an RC structure filled with masonry, and the performance of the dynamic characteristics. A comparison was performed with the readings provided by a reference accelerometer. Figueiredo et al. [16] used the *App4SHM* smartphone app to detect anomalies in two structures: (i) laboratory steel beams and (ii) two twin post-tensioned concrete bridges. The app collected vibration data from the structures using the device's internal sensors and then estimated the structure's natural frequencies. Finally, damage assessment was carried out by comparing the acquired data with those obtained in the reference scenario.

Similarly, Sun W. [17] proposed using smartphone-based instrumentation to monitor the structural integrity of walls in modern buildings. In this case, the device plays the role of both actuator and sensor. The structure is excited by the smartphone vibrating at a certain frequency, and the dynamic responses are captured by its internal accelerometer and gyroscope sensors. This instrumentation strategy provides flexibility and cost-effectiveness in real-world monitoring cases. Nowadays, smartphones are equipped with dynamic sensors and can also play the role of actuators. In addition, as evaluated in [17], the model of smartphone used, as well as aspects related to the way and direction in which the devices are held during monitoring, did not significantly affect the system's performance.

To fulfill the objective of using dynamic signals collected from smartphones to monitor the integrity of walls, Sun W. [17] used Z-axis data from both the device's accelerometer and gyroscope. Statistical features and Mel-Frequency Cepstral Coefficients (MFCC) [18] were extracted from the signals and submitted to Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN) models [19] to predict the structural condition of the walls. As a result, combining MFCC with KNN made it possible to achieve an accuracy level of around 99.20%.

Despite the good results obtained by Sun W. [17], when evaluating the models using only the accelerometer or gyroscope signals separately, the method used failed to achieve a similar performance. As a consequence, the accuracy decreased to around 75%. Thus, considering that the use of gyroscopes in structural monitoring is an uncommon approach and that the evaluation of accelerometer signals is a robust and widely used approach in the field of vibration-based SHM, this work aims to assess the presence of wall damage using only accelerometer readings. In addition, we are interested in proposing a computational approach that improves the results of wall damage detection only by exploiting other features of the signals, keeping the same three machine learning (ML) models as a comparison.

The main contributions of this work are based on two aspects: the first consists in exploiting features from the spectral domain, while the second relies in adopting wall material information as an input variable. In addition, as new variables are introduced into the computational approach, the grid search (GS) process for optimizing the hyperparameters of the models is adopted.

The computational experiments are carried as in [17] and are evaluated using 5 performance metrics: accuracy, ROC-AUC, Jaccard, precision, and recall. Initially, experiments were conducted to assess the effectiveness of the suggested spectral features concerning those previously evaluated by Sun W. [17]. Subsequently, the computational method was modified to include information on the wall's constituent material as an input variable for the models, along with each set of features evaluated. Finally, the grid search process was introduced into the modified method to optimize the set of internal parameters of the models.

2. Material and Methods

This section details the dataset used and presents an overview of the machine learning models applied to solve the problem.

2.1 Dataset

Sun W. [17] deployed an application (Vibwall) for smartphones to record accelerometer and gyroscope data while activating the device's vibrator. The vibrator was configured to serve as the stimulus at five distinct frequencies (10Hz, 20Hz, 30Hz, 40Hz, and 50Hz). The experiments were carried out on walls of different materials, including concrete, wood, and brick, from modern building environments such as apartments and offices, as illustrated in Figure 1. To make it easier to identify the samples, the experimental tests were conducted so that detectable anomalies were visible using visual inspection methods, such as human observation or camera systems.

A total of 3,000 measurements were collected, half from walls with cracks and the other half from intact walls. In other words, for each type of wall, 500 measurements were taken in healthy structures and another 500 in abnormal conditions, considering only the accelerometer response readings over the z-axis (out of the wall's plane). In addition, the sensor was configured to sample data at a rate of 100 Hz. Each vibration stimulus was applied for approximately 5 seconds, although an analysis of the data revealed 24 signals with shorter durations and signals with longer durations than the established 5 seconds.

2.2 K-Nearest Neighbors

This ML algorithm classifies an unlabeled test sample based on the majority class of its nearest neighbors. This process involves calculating the distances between the test and training samples using a specific distance measure [20].

The KNN is recognized for its simplicity and effectiveness in solving various classification problems, showcasing its adaptability to different types of data. Its simplicity lies in the fact that the model does not require an explicit training process and is considered a lazy learning method. This means that the training phase is practically non-existent, as KNN simply stores the training data and only carries out intensive calculations during the classification phase [21].

However, it is important to acknowledge its drawbacks, such as the challenge of determining the optimal k value, which represents the number of neighbours to be considered. Very small values of k can result in a model that is sensitive to noise in the data, while very large values can dilute the influence of relevant near neighbours [22]. The high computational load is another challenge, as KNN requires calculating the distances of each test sample from all the training samples, which can be unfeasible for large data sets. In addition, KNN requires a substantial amount of memory to store all the training data, which can be limiting in large-scale applications.

Various strategies have been proposed to adapt to these challenges, such as using inverted indexes or ensemble learning to find the optimal k, reducing the size of the training dataset, or employing approximate KNN classification methods [23].

The operation of the KNN algorithm is based on two main phases. During training, the samples and their class labels are stored, ensuring no missing or non-numeric data. In the classification phase, each test sample is classified based on a majority vote among its nearest neighbours. Distances between the test sample and all training samples are calculated using a predefined distance function, such as the Euclidean distance, and the class most frequently occurring among the k nearest neighbors is assigned to the test sample. This approach ensures that the test sample is classified accurately based on the most relevant and proximate training data [22]. In addition, it is also possible to define a weight function used for prediction, such as uniform weight, in which all the points in each neighbourhood are weighted equally, and weight by distance, in which the weight points are weighted by the inverse of their distance [24].

2.3 Decision Tree Classifier

The primary objective of DT is to construct a model that predicts the target variable by inferring simple decision rules from the data features, effectively functioning as piecewise constant approximations. Each internal node of the tree represents a test on a feature, each branch corresponds to the outcome of the test, and each leaf node represents a predicted value of the target variable. This structure allows DTs to approximate functions like a sine curve through a series of if-then-else rules, where deeper trees yield more complex regulations and a more accurately fitted model [25].

In addition, computer implementations of DT, such as those available in scikit-learn, make it possible to adjust the behaviour of the model using some additional parameters, such as 'splitter' and 'criterion'. The first of these is a parameter related to the strategy used to choose the split at each node; for example, selecting the best split at each node (best) or selecting a random subset of resources and then the best split from that subset (random). The *criterion* determines the function used to measure the quality of a split [26]. These options offer flexibility to adapt the tree to specific data sets and tasks, which can improve performance and generalisation.

One of the key strengths of decision trees is their versatility, making them a reassuring choice for a wide range of data analysis tasks. Their intuitive visual representation makes it easy to understanding and interpret, allowing for greater comprehension of the decision-making process without the need for extensive technical knowledge. Unlike other ML methods, DT requires minimal data preparation, eliminating the need for extensive data normalization and handling of missing values. They can handle both numerical and categorical data efficiently, making them applicable in diverse contexts. Furthermore, the computational cost of making predictions with a DT is logarithmic relative to the number of training data points, ensuring efficiency even with large datasets. Decision trees also support multi-output problems, and their transparency as a white-box model ensures that their decision-making process is fully interpretable. Statistical tests can validate these models, ensuring reliability even when un-

Figure 1. The wall's structural health detection across the concrete wall, wooden wall, and brick walls in modern buildings (adapted from [17]).

derlying assumptions are partially violated [27].

Despite these advantages, DT exhibits some limitations. They tend to overfit the data, especially when the tree is deep, leading to poor generalization to unseen data. Another significant drawback is their instability; small variations in the data can lead to substantially different trees, making them sensitive to noise. The piecewise constant nature of DT limits their performance in tasks that require extrapolation beyond the range of training data. Furthermore, learning an optimal DT is NP-complete problem, meaning that exact solutions are computationally infeasible for large datasets. As a consequence, heuristic algorithms, such as the greedy algorithm, are commonly employed, which makes decisions at each node based on the best local option rather than a global optimum. Moreover, DT struggles with complex patterns like XOR (exclusive OR) and can produce biased trees if some classes dominate the dataset, highlighting the importance of balancing the dataset before training [25].

2.4 Random Forest Classifier

One way to reduce the risk of overfitting is by constructing an ensemble of decision trees, as in the Random Forest method. This ensemble combines predictions from several classifiers using a majority vote decision rule to improve the robustness and generalisability of the model [19, 28].

RF comprises multiple DT, in which each individual tree is built from a random sample of the data set, using the bagging technique (bootstrap aggregating). In this process, different subsets of the training set are randomly generated with replacement, creating diversity among the trees in the model [28]. This randomness is fundamental to reducing the correlation between the trees and, consequently, reducing the variance of the model.

The trees within a Random Forest are developed using random subsets of the available features, selected at each node of the tree. This process introduces even more variation between the trees, increasing the model's resilience against overfitting. Each tree votes independently for the most frequent class to classify a new input vector. The class that gets the most votes among all the trees in the forest is the classifier's final decision for a data instance [29, 30].

To construct a decision tree within Random Forest, it is essential to choose an appropriate measure of feature importance, such as Gini impurity or entropy, to guide the splitting of nodes. In addition, the number of trees to be generated, the maximum depth of the trees and the number of features to be considered in each split are also parameters that influence the model's performance. Careful choice of these parameters can improve the Random Forest's ability to generalise, especially in complex or high-dimensional data sets. Finally, the forest classifies each new data instance by passing it through all the trees, selecting the class with the majority of votes [30].

3. Proposed Approach

The main improvements proposed are structured around two main aspects. The first one consists in exploring and evaluating the performance of the models from the perspective of 12 spectral features. Subsequently, information about the wall's constituent material is used as an input variable. In addition, once a new input variable has been introduced, a process of optimizing the internal parameters of the models was carried out via grid search. In this case, there is also an interest in finding a set of hyperparameters that best fit the models from the perspective of spectral features.

3.1 Signal Analysis

The structure for reading dynamic signals in the original work established a standard duration of 5 seconds, i.e. 500 points. However, preliminary data analysis revealed some signals of the original dataset whose duration exceeded 5 seconds and other signals shorter than the established standard. In the first case, the problem was dealt with by ignoring the extra portion of the signals without any significant loss of information. Regarding the second one, the data analysis revealed 24 files (0.8% of the database size) containing signals lasting less than the established 5 seconds. Thus, considering that the occurrence of signals with shorter duration was infrequent, it was decided to discard such signals from the database without significant loss of information. As a result, the data set used in this study consists of 2976 vibration signals, and no imbalance problems were generated with this approach.

3.2 Data Processing

The problem of damage detection in the modern building walls was evaluated in [17] using 9 manually implemented statistical features described in Table 1, and MFCCs extracted from the signals via the *python_speech_features* library [31].

Table 1. Statistical features extracted from the signals.

Feature	Description
X_1	Mean Absolute Value (MAV).
X_2	Variance (VAR).
X_3	Root Mean Square (RMS).
X_4	Standard Deviation (STD).
X_5	Mean Absolute Deviation (MAD).
X_6	Skewness of the frequency domain.
X_7	Kurtosis of the frequency domain.
X_8	Interquartile (IQR) range.
X_9	Energy of the signal.

We propose here using a set of spectral features, as described in Table 2, in the context of wall damage detection via smartphone's accelerometer signals. Also, we propose inserting information of the wall's constituent material as input variables. The hypothesis behind providing this information to the models is that different materials show different vibration responses due to their physical and mechanical properties [32]. Therefore, using material identification may improve the success rate in detecting anomalies in the wall.

Finally, in both approaches, the feature sets were normalized in $[0,1]$ linearly.

Table 2. Proposed spectral features set, which were extracted from the signals via TSFEL [33].

Feature	Description
S_1	Spectral Centroid
S_2	Spectral Amplitude Decrease
S_3	Spectral Distance
S_4	Spectral Entropy
S_5	Spectral Kurtosis
S_6	Spectral Roll-off
S_7	Spectral Skewness
S_8	Spectral Spread
S_9	Maximum Power Spectrum Density
S_{10}	Maximum Frequency
S_{11}	Median Frequency
S_{12}	Power Spectrum Density Bandwidth

3.3 Grid Search for Internal Parameters

After introducing the information on the wall's constituent material as an input variable and proposing a set of spectral feature data to be evaluated, it is important to carry out a new evaluation of the models' internal parameters. Therefore, a hyperparameter optimisation approach based on Grid Search combined with Stratified Cross Validation was used. In addition, this type of approach is a way of ensuring that machine learning models adapt to possible different monitoring scenarios.

The process took into account a Stratified Cross Validation (SCV) with 5 divisions (cv=5), carried out in 30 independent iterations. SCV was chosen to ensure that each of the splits maintains the original proportion of classes, which is crucial in unbalanced datasets. For each iteration, a new set of training and test divisions is generated, ensuring robustness in the results. The training set is then further split into training and validation using the *train_test_split* method with 25% of the data reserved for validation. This stage is essential for evaluating the performance of the different hyperparameters during the Grid Search process. The details of the candidate solutions for the internal parameters are shown in Table 3.

Table 3. Hyperparameter candidates for the grid search procedure.

Classifier	Hyperparameter	Candidates
K-Neighbors	$n\_neighbors$	$[1,2,3,5,10,50,100]$
	$weights$	$[uniform, distance]$
Decision Tree	max_depth	$[2,5,10,15]$
	$criterion$	$[gini, entropy, log\_loss]$
	$splitter$	$[best, random]$
Random Forest	max_depth	$[2,5,10,15]$
	$n\_estimators$	$[10,50,100,200]$
	$criterion$	$[gini, entropy, log\_loss]$

The Grid Search process was used to exhaustively explore all possible combinations of the specified hyperparameters. For each combination, the model is trained on the training set and evaluated on the validation set. After testing all the combinations, the combination that results in the best performance on the validation set is selected as the best. This process is repeated for each iteration of cross-validation, and the best hyperparameters found in each iteration are stored for later analysis.

It is worth noting that the computational experiments conducted without the GS optimization process considered the same internal parameters as the experiments carried out by [17]. Thus, $max\_depth = 2$ and $n\_estimators = 10$ were considered for the Random Forest model, $K = 2$ for the k-neighbors classifier and, finally, the Decision Tree model tuning was kept to the implementation standards of the scikit-learn library [26].

4. Computational Experiments

The database used in this research was kindly made available by Wei S. upon request. With the exception of these data, all other materials necessary for the reproduction of this work can also be made available upon request.

The developed codes were implemented in Python and the feature extraction process was based on libraries numpy [34] (version 1.23.5) and scipy [35] (version 1.9.3) for statistical ones, *python_speech_features* [31] (version 0.6) for extracting MFCCs and TSFEL [33] (version 0.1.4) to retrieve spectral information from the dynamical signals. Additionally, the computational experiments were conducted on a computer with the following specifications: Intel(R) Core(TM) i5-1135G7 CPU @ 2.40GHz, 8 GB RAM, and Windows 10 Home as an operational system.

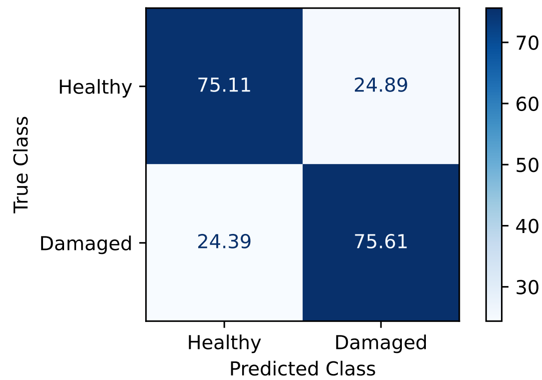
To assess the model's performance comprehensively, the computational experiments were conducted 30 times using different random data splits via a 5-fold stratified cross-validation. It resulted in a total of 150 model evaluations across various dataset partitions and from the perspective of 5 performance metrics: accuracy, ROC-AUC, Jaccard, precision, and recall.

Table 4 shows the results achieved by the original method implemented and by the modified method with the first proposed changes, i.e. with the addition of the input variable, and finally by the optimized proposed method.

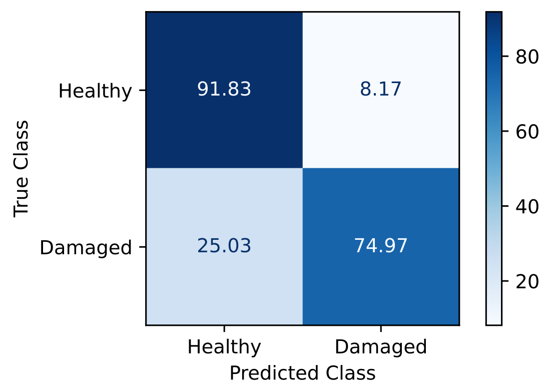
When evaluating the anomaly detection capacity of the original method - without considering the spectral features, it is possible to see that the best average accuracy is about 75%, as was expected, when considering the statistical features and DT. However, KNN with the same feature set obtained similar results in all metrics, even showing a better average precision score. Furthermore, the superiority of the spectral features suggested here could be observed. The KNN model achieved an average accuracy of 83.99%, which corresponds to an increase of 8.63%. After introducing the proposed modifications but without considering the optimization process of the models yet, it can be seen that it was possible to raise the classification performance even further. In this case, an average accuracy of 95.58% was achieved using the set of spectral features and the KNN classifier. In addition, the fact of simply indicating the main constituent material of the wall led to an improvement in the classification performance of the models, with an average increase in accuracy of up to 17.04% (60.30% to 77.34%).

A significant performance enhancement resulting from model optimization can also be observed, mainly when considering the RF classifier and the signal characteristics from statistical and spectral domains. In these cases, an increase of 23.39% (68.01% to 91.40%) and 21, 96% (70.72% to 92.68%) was observed in the average accuracy of the respective feature sets. The model's improvement was so remarkable that, when combined with the statistical features, its performance exceeded that of the DT model when applied to the set of spectral attributes. Still in this sense, it can be noted that in general, except for the set of MFCCs, the optimized RF and KNN models performed better than the DT. It can also be seen that the best performances in each case are related to the KNN classifier.

As a complementary analysis, Figures 2 and 3 shows the average confusion matrices of the best models of both the original framework and when adopting the suggested improvements respectively. Returning to the classification performances presented in Table 4, when considering the original method, the usage of spectral features leads to an increase in the average performance of the models. However, analyzing the confusion matrices related to the best results of this method (Figures 2a and 2b) reveals that, even though a reduction in the error of identifying healthy structures was reduced from 24.89% to 8.17%, a subtle increase in the classification error of signals from damaged structures was observed (24.39% to 25.03%). Similarly, optimizing the hyperparameter of KNN improved the quality of the solutions, albeit in a limited way. However, an evaluation of the confusion matrices (Figures 3a and 3b) reveals a change in the behaviour of the classification mistakes. Thus, after the optimization process, there was an increase in the misclassification rate of signals from healthy structures and a reduction in the proportion of mistakes classification regarding signals from damaged structures.



(a) Statistical and DT



(b) Spectral and KNN

Figure 2. Confusion matrices from the best cases of the original method.

After verifying that the models were effective in detecting anomalies when considering the statistical and spectral perspectives, we generated models using a set of features formed by combining them. Therefore, a new set was constructed containing 21 features from the signals (9 statistical + 12 spectral) in addition to the information relating to the wall materials. The results achieved in this case are shown in Table 5. When evaluating them, it can be seen that the combination of these sets of features led to an increase in the performance of the models in the three methods analyzed: original, modified, and optimized. In this way, it was possible to increase the average accuracy to about 97.78%.

Complementary, the highest average misclassification rate drops from 4.57% (value of false positives in Figure 3b) to 2.73%, in the latter case being associated with signals from damaged structures (false negatives), as can be seen in Fig-

Table 4. Average performance metrics and their standard deviation (%) achieved by using the original method and the modified ones. The bold value indicated the highest average performance achieved in each method evaluated. The underlined ones highlight the best performance by considering the reference paper approach.

Method	Feature Set	ML Model	Performance Metrics Accuracy	ROC-AUC	Jaccard	Precision	Recall
Original	Statistical	KNN	74.69 (1.54)	74.67 (1.54)	55.02 (2.41)	<u>76.40 (1.85)</u>	74.67 (2.28)
		DT	<u>75.36 (1.68)</u>	<u>75.36 (1.68)</u>	<u>60.52 (2.20)</u>	75.39 (1.89)	<u>75.36 (2.39)</u>
		RF	68.06 (1.82)	68.08 (1.82)	55.70 (1.90)	69.39 (2.50)	68.08 (4.15)
	MFCC	KNN	56.31 (1.81)	56.26 (81)	26.88 (4.26)	58.44 (2.86)	56.26 (7.12)
		DT	60.30 (2.07)	60.30 (2.07)	42.99 (2.39)	60.31 (2.12)	60.30 (2.97)
		RF	62.17 (1.99)	62.18 (1.99)	47.11 (2.42)	62.37 (2.20)	62.18 (3.54)
	Spectral	KNN	**83.99 (1.43)**	**83.98 (1.43)**	**70.37 (2.42)**	**84.88 (1.68)**	**83.98 (1.95)**
		DT	81.31 (1.56)	81.31 (1.56)	68.54 (2.17)	81.35 (1.88)	81.31 (2.28)
		RF	70.66 (1.78)	70.69 (1.78)	59.37 (1.98)	72.98 (2.70)	70.69 (4.37)
Modified	Statistical	KNN	89.66 (1.02)	89.65 (1.02)	80.23 (1.88)	90.18 (1.40)	89.65 (1.61)
		DT	89.04 (1.18)	89.04 (1.18)	80.26 (1.92)	89.06 (1.53)	89.04 (1.71)
		RF	68.01 (1.66)	68.03 (1.66)	55.43 (1.77)	69.24 (2.45)	68.03 (4.65)
	MFCC	KNN	56.31 (1.75)	56.26 (1.74)	26.80 (4.28)	58.48 (2.83)	56.26 (7.28)
		DT	77.34 (1.75)	77.34 (1.75)	62.97 (2.42)	77.37 (2.03)	77.34 (2.53)
		RF	62.11 (2.10)	62.13 (2.10)	47.01 (2.45)	62.30 (2.28)	62.13 (3.54)
	Spectral	KNN	**95.58 (0.82)**	**95.58 (0.82)**	**91.37 (1.56)**	**95.67 (1.10)**	**95.58 (1.17)**
		DT	89.12 (1.80)	89.12 (1.80)	80.43 (2.92)	89.15 (2.16)	89.12 (2.36)
		RF	70.72 (1.89)	70.75 (1.89)	59.37 (2.04)	72.92 (2.62)	70.75 (4.06)
Optimized Proposed	Statistical	KNN	90.45 (1.06)	90.45 (1.06)	82.45 (1.84)	90.51 (1.82)	90.45 (2.11)
		DT	88.92 (1.38)	88.92 (1.38)	80.00 (2.28)	88.98 (2.04)	88.92 (2.40)
		RF	91.40 (1.05)	91.40 (1.05)	84.12 (1.79)	91.42 (1.49)	91.40 (1.66)
	MFCC	KNN	55.91 (2.28)	55.91 (2.27)	37.34 (8.89)	57.17 (4.18)	55.91 (20.43)
		DT	78.87 (2.86)	78.87 (2.86)	64.19 (4.00)	79.21 (3.87)	78.87 (5.18)
		RF	67.49 (1.97)	67.50 (1.97)	52.97 (2.38)	67.78 (2.18)	67.50 (3.06)
	Spectral	KNN	**95.87 (0.84)**	**95.87 (0.84)**	**92.09 (1.54)**	**95.89 (1.29)**	**95.87 (1.37)**
		DT	90.24 (2.29)	90.24 (2.29)	82.30 (3.77)	90.28 (2.63)	90.24 (2.85)
		RF	92.68 (1.16)	92.68 (1.16)	86.38 (1.99)	92.70 (1.56)	92.68 (1.68)

ure 4.

4.1 Parametric Analysis

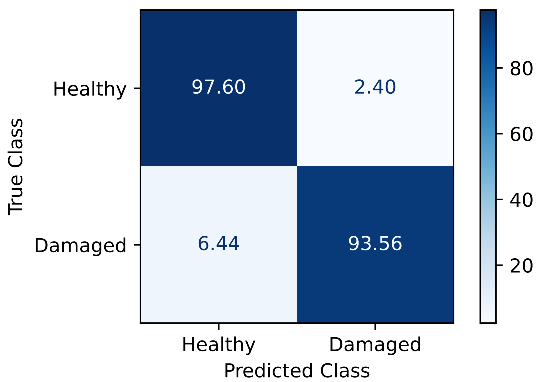
The best classification performance achieved in this paper was found when using the KNN model combined with the concatenated set of statistical and spectral features. In this case, a parametric analysis conducted after the optimization process via GS revealed that the parameters $n\_neighbors = 1$ and $weights = uniform$ were the configurations mostly selected as the best, respectively in 75 and 131 of the 150 model runs, as can be seen in Figure 5. A similar behaviour of the internal parameters was found in the evaluation of the best case when only spectral attributes were considered. In this case, the same best configurations were found ($n\_neighbors = 1$ and $weights = uniform$), but with a respective selection frequency of 95/150 and 126/150. Thus, it can be noted that, among the possible configurations for *n_neighbors*, the grid search selections were more concentrated on the lower values of this parameter, revealing a better fit of the KNN model when using a smaller number of neighbors for class prediction.

Parametric and feature importance analyses were also carried out for DT and RF. Firstly, this analysis revealed a difficulty in defining the best model parameters, as the frequency distribution was very similar between some of the candidate solutions, as can be seen in Figures 6 and 7. The most obvious case is related to the *criterion* parameter of the DT model, whose selection frequency is equally distributed between the *gini* and *entropy* candidates. On the other hand, the *n_estimators* parameter for the RF model showed the best convergence in the selection, so the configuration comprising 15 estimators was selected in 147 of the 150 iterations. In general, however, it can be said that the other best parameter settings were $max\_depth = 15$, $Splitter = best$ for DT, and $criterion = entropy$, $n\_estimators = 200$ for RF.

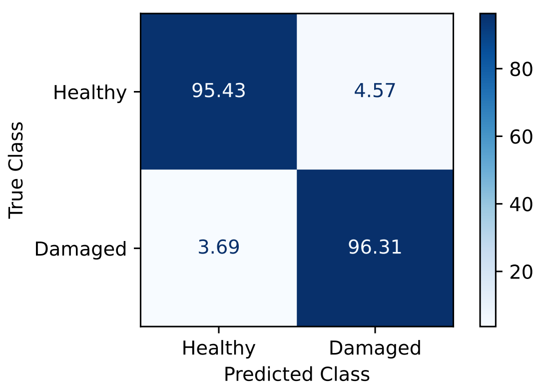
Concerning the cumulative relevance of the features, Figure 8 shows the importance of each feature in the decision-making process of two machine learning models: Decision Tree and Random Forest. The height of each bar represents the contribution of a given attribute to the model's performance. The higher the bar, the more important the feature is for the model in question. When analysing the bar chart, the importance of the variable containing information on the wall's constituent material is noticeably higher in both mod-

Table 5. Average performance metrics and its standard deviation (%) achieved by considering the concatenated statistical and spectral features set. The bold value indicated the highest average performance achieved in each method evaluated.

Method	ML Model	Performance Metrics Accuracy	ROC-AUC	Jaccard	Precision	Recall
Original	KNN	**87.90 (1.27)**	**87.90 (1.27)**	**77.20 (2.26)**	**88.48 (1.53)**	**87.90 (1.77)**
	DT	83.57 (1.65)	83.57 (1.65)	71.87 (2.45)	83.60 (1.97)	83.57 (2.28)
	RF	72.05 (1.80)	72.05 (1.80)	60.76 (1.91)	74.22 (2.60)	72.05 (4.05)
Modified	KNN	**97.59 (0.59)**	**97.58 (0.59)**	**95.20 (1.17)**	**97.64 (0.77)**	**97.58 (0.81)**
	DT	94.29 (1.14)	94.29 (1.14)	89.21 (2.05)	94.31 (1.47)	94.29 (1.55)
	RF	72.24 (2.09)	72.27 (2.09)	61.03 (2.08)	74.49 (2.65)	72.27 (3.88)
Optimized Proposed	KNN	**97.78 (0.62)**	**97.77 (0.62)**	**95.62 (1.21)**	**97.79 (0.93)**	**97.77 (0.96)**
	DT	94.08 (1.46)	94.08 (1.46)	88.81 (2.55)	94.11 (1.85)	94.08 (1.96)
	RF	95.58 (1.02)	95.58 (1.02)	91.44 (1.91)	95.62 (1.30)	95.58 (1.36)



(a) Spectral and KNN



(b) Spectral and KNN

Figure 3. Confusion matrices from the best cases of the proposed method without (a) and with (b) hyperparameter optimization.

els. The following 6 features are also more prominent than the other ones for both classifiers: MAV, Frequency Skewness, Frequency Kurtosis, Energy, RMS, and Spectral Spread. Specifically analysing the importance of the variables for each model, it can be seen that the DT model tends to focus on a smaller number of highly informative attributes, which can be an advantage in terms of interpretability. For the Random Forest model, on the other hand, a more uniform distribution

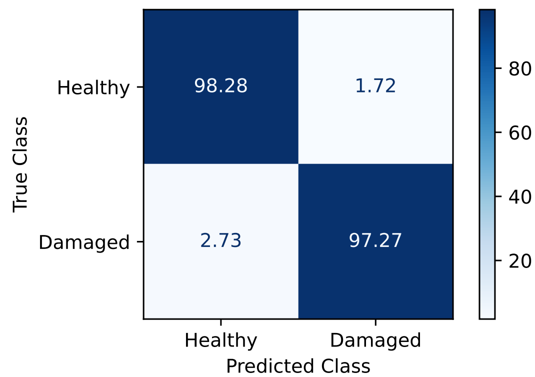


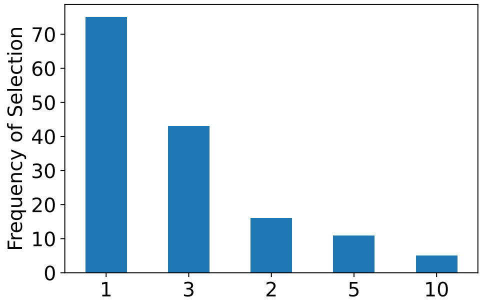
Figure 4. Confusion matrices of the best cases when considering the optimized proposed method and the set composed of the statistical and spectral features.

of importance among the other attributes is observed, indicating that the model considers a wider range of characteristics when making its decisions.

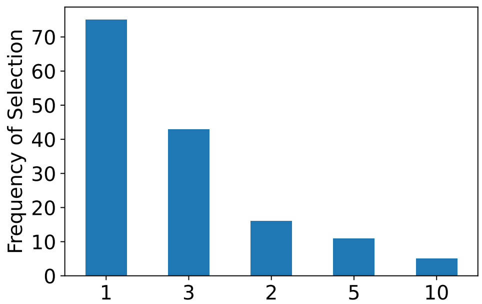
4.2 Overall Discussion

Analysing the results, it is clear that the three main modifications proposed have had a positive effect, increasing the assertiveness rate of the models. In particular, optimising the models' hyperparameters combined with cross-validation enabled the models to perform better, especially the RF. This indicates that the proposed model could mitigate possible overfitting effects. Still, in this sense, it was possible to achieve an average accuracy (97.78%) as good as that obtained in the best case of the original article (99.20%), with the difference that the approach used in this work takes only the accelerometer signals perspective, without introducing the information collected by the gyroscope.

It was also possible to observe that, in general, the set formed by the MFCCs was associated with the lowest averages of the performance metrics evaluated. This is probably due to the inefficient configuration of the library method used to extract them, which generated a feature matrix made up of approximately 6460 columns, while the other sets were made up of 9 (statistical) and 12 (spectral) attributes. As an alterna-



(a) Parameter: *n_neighbors*



(b) Parameter: weights.

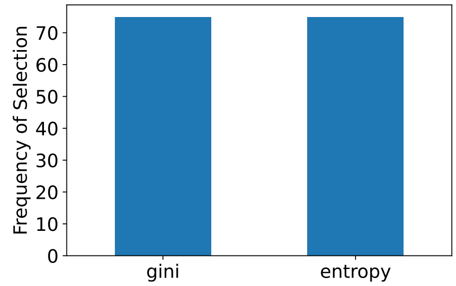
Figure 5. Best internal parameters identified by the grid search process after executing KNN with the statistical and spectral combined features set.

tive to be evaluated in future work to remedy this problem of the hyperdimensionality of the MFCC, it is suggested that a more simplified MFCC extraction method be applied, such as that used in [36].

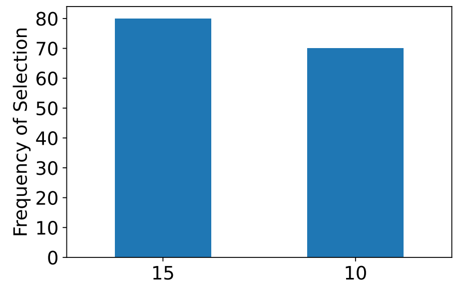
When evaluating the best set of features (a combination of statistical and spectral characteristics), it can be seen that the 7 attributes that stood out were the same for DT and RF. Furthermore, 5 out of them come from the statistical set, 1 from the spectral set, and the last one consists of wall material information, which was found to be the most relevant feature in both classifiers. It is therefore clear that the proposed approach was able to contribute to progress in the problem of detecting wall damage from smartphone accelerometer signals.

4.3 Strengths and Limitations

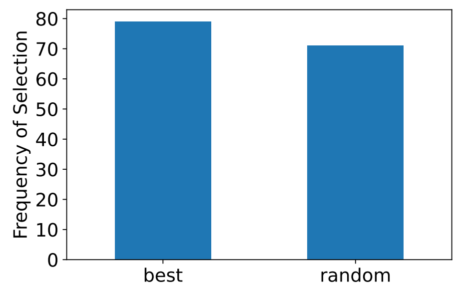
Various approaches have been explored for detecting structural damage in walls, most notably non-destructive monitoring methods. However, these methods tend to be costly and sensitive to environmental changes such as lighting and ambient noise. In this sense, the proposal in this study differs from these approaches by employing affordable sensors, such as accelerometers built into smartphones, combined with machine learning techniques to analyse spectral data and material information. This combination offers significant potential for reducing costs and facilitating data collection in the field, highlighting the feasibility of large-scale applications. Furthermore, to the best of the authors' knowledge, there is no



(a) Criterion.



(b) *max_depth*.



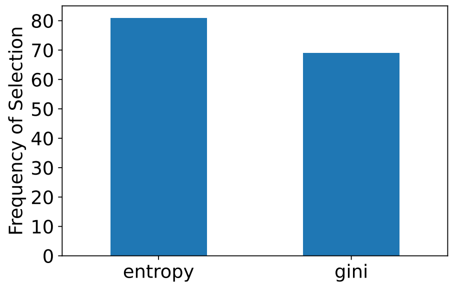
(c) Splitter.

Figure 6. Best internal parameters identified by the grid search process for DT model in the set formed by the statistical and spectral features.

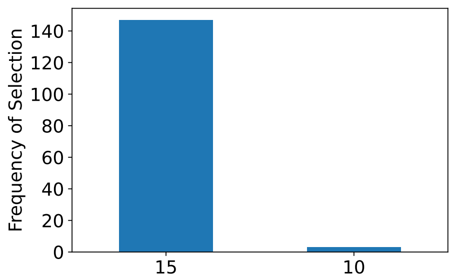
other record, apart from the reference paper [17], of this type of technique being used to monitor walls.

When considering applications in real conditions, even though the reference paper [17] carried out tests in this scenario and sought to assess the effect of different data collection conditions, it is possible that materials other than those already analysed could be subjected to the monitoring process. In this case, classification performance could be affected. In this respect, even though this work has obtained promising results and outlined strategies to guarantee a level of generality in the model, dealing with a different scenario could still be a problem. For this reason, we suggest that future efforts be made to increase the model's degree of generalisation by acquiring vibration data from different types of walls and materials in real situations. Furthermore, we suggest that these new acquisitions seek to vary the data collection conditions.

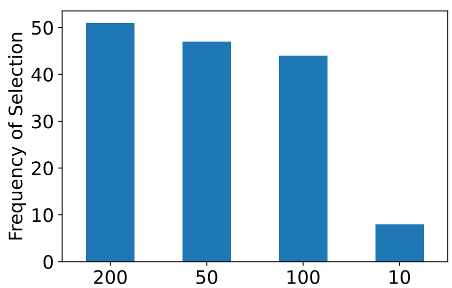
In addition, the integration of the proposed models into mobile applications could facilitate the automation of the detection process, allowing users to carry out real-time as-



(a) Criterion.



(b) *max_depth*.



(c) *n_estimators*.

Figure 7. Best internal parameters identified by the grid search process for RF model in the set formed by the statistical and spectral features.

sessments in a practical and efficient manner. Combined with this, the nature of the problem and the solution developed motivate the development of a federated learning structure to detect anomalies in the walls of modern buildings, as well as transfer learning techniques to adjust predictive models to different scenarios, making them more robust to regional and environmental variations.

Finally, the authors also encourage evaluating more complex ML models, such as artificial neural networks, due to their greater pattern recognition capacity, including those that can act directly on the raw vibration data. Furthermore, in this scenario of using complex ML models, it is strongly suggested the usage of more robust hyperparameters optimisation techniques, such as differential evolution [37] or other bio-inspired algorithm.

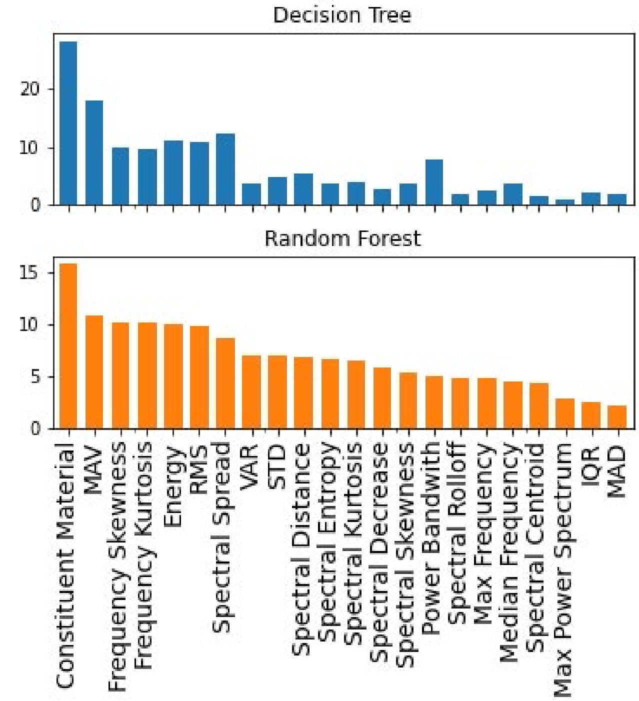


Figure 8. Accumulated importance of the features during the 150 runs of the model.

5. Concluding Remarks

We proposed here the use of spectral features and wall material information as features for ML models in the context of damage detection in modern building walls using only smartphone accelerometer data. In addition, due to the introduction of these features, a grid search process was also carried out to optimize the hyperparameters of the models. The proposed contributions lead to an increase in average performance in terms of accuracy by approximately 23%.

Although the best scenario found for detecting damage was with the combination of statistical and spectral information from the signals, the evaluation of the set of spectral features showed that these characteristics extracted from the signals were sufficient to increase the performance of the models evaluated. A similar behaviour was seen after introducing wall material information as an input variable. The process of optimizing the internal parameters of the models also showed its contribution to the increase in average accuracy, especially in the case of the RF classifier, where the highest increases were observed. For the DT and KNN models, this process showed less significant improvements.

Acknowledgements

The authors acknowledge the Brazilian funding agency CAPES - Finance Code 001 for their financial support. Also, we thank the support provided by CNPq, FAPEMIG, and UFJF.

Author contributions

Tales Boratto: Conceptualization, methodology, software, writing - original draft, writing - review and editing. Douglas Lima Fonseca: Software, writing - original draft, writing - review and editing. Heder Soares Bernardino: Conceptualization, methodology, writing - review and editing, and supervision. Alex Borges: Conceptualization, methodology, and writing - review and editing. Alexandre Cury: Resources, validation, and writing - review and editing. Leonardo Goliatt: Methodology, supervision, and writing - review and editing.

References

[1] FRANGOPOL, D. M.; LIU, M. Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost. *Structures and infrastructure systems*, Routledge, p. 96–108, 2019.

[2] ROBSON, P. *Structural repair of traditional buildings*. [S.l.]: Routledge, 2015.

[3] STEWART, M. G.; ROSOWSKY, D. V. Structural safety and serviceability of concrete bridges subject to corrosion. *Journal of Infrastructure systems*, American Society of Civil Engineers, v. 4, n. 4, p. 146–155, 1998.

[4] ALVES, V.; CURY, A.; CREMONA, C. On the use of symbolic vibration data for robust structural health monitoring. *Proceedings of the Institution of Civil Engineers - Structures and Buildings*, v. 169, n. 9, p. 715–723, 2016.

[5] ALVES, V.; CURY, A. An automated vibration-based structural damage localization strategy using filter-type feature selection. *Mechanical Systems and Signal Processing*, v. 190, p. 110145, 2023. ISSN 0888-3270.

[6] AZIMI, M.; ESLAMLOU, A. D.; PEKCAN, G. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, v. 20, n. 10, 2020. ISSN 1424-8220.

[7] SARMADI, H. et al. Review on smartphone sensing technology for structural health monitoring. *Measurement*, v. 223, p. 113716, 2023. ISSN 0263-2241.

[8] FINOTTI, R. P. et al. Numerical and experimental evaluation of structural changes using sparse auto-encoders and svm applied to dynamic responses. *Applied Sciences*, v. 11, n. 24, 2021. ISSN 2076-3417.

[9] MAGALHãES, F.; CUNHA, A.; CAETANO, E. Vibration based structural health monitoring of an arch bridge: From automated oma to damage detection. *Mechanical Systems and Signal Processing*, v. 28, p. 212–228, 2012. ISSN 0888-3270. Interdisciplinary and Integration Aspects in Structural Health Monitoring.

[10] GENTILE, C.; RUCCOLO, A.; CANALI, F. Continuous monitoring of the milan cathedral: dynamic characteristics and vibration-based shm. *Journal of Civil Structural Health Monitoring*, Springer Science and Business Media LLC, v. 9, n. 5, p. 671–688, out. 2019. ISSN 2190-5479.

[11] UBERTINI, F.; COMANDUCCI, G.; CAVALAGLI, N. Vibration-based structural health monitoring of a historic bell-tower using output-only measurements and multivariate statistical analysis. *Structural Health Monitoring*, v. 15, n. 4, p. 438–457, 2016.

[12] GHOSHAL, A. et al. Structural health monitoring techniques for wind turbine blades. *Journal of Wind Engineering and Industrial Aerodynamics*, v. 85, n. 3, p. 309–324, 2000. ISSN 0167-6105.

[13] PEREIRA, S. et al. Vibration-based damage detection of a concrete arch dam. *Engineering Structures*, v. 235, p. 112032, 2021. ISSN 0141-0296.

[14] FENG, M. et al. Citizen sensors for shm: Use of accelerometer data from smartphones. *Sensors*, v. 15, n. 2, p. 2980–2998, 2015. ISSN 1424-8220.

[15] KANG, J.-D.; BAEK, E.-R.; PARK, S.-H. Evaluation of vibration detection using smartphones in a two-story masonry-infilled rc frame building. *Buildings*, v. 13, n. 4, 2023. ISSN 2075-5309.

[16] FIGUEIREDO, E. et al. Smartphone application for structural health monitoring of bridges. *Sensors*, v. 22, n. 21, 2022. ISSN 1424-8220.

[17] SUN, W. Vibwall: Smartphone's vibration challenge-response for wall crack detection. *ACM J. Comput. Sustain. Soc.*, Association for Computing Machinery, New York, NY, USA, v. 1, n. 1, sep 2023.

[18] ABDUL, Z. K.; AL-TALABANI, A. K. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, v. 10, p. 122136–122158, 2022.

[19] SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014. ISBN 1107057132.

[20] LIAO, Y.; VEMURI, V. R. Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, Elsevier, v. 21, n. 5, p. 439–448, 2002.

[21] GUO, G. et al. Knn model-based approach in classification. In: MEERSMAN, R.; TARI, Z.; SCHMIDT, D. C. (Ed.). *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 986–996. ISBN 978-3-540-39964-3.

[22] ALFEILAT, H. A. A. et al. Effects of distance measure choice on k-nearest neighbor classifier performance: A review. *Big Data*, Mary Ann Liebert Inc, v. 7, n. 4, p. 221–248, dez. 2019. ISSN 2167-647X.

[23] LAAKSONEN, J.; OJA, E. Classification with learning k-nearest neighbors. In: IEEE. *Proceedings of international conference on neural networks (ICNN'96)*. [S.l.], 1996. v. 3, p. 1480–1483.

[24] DUDANI, S. A. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6, n. 4, p. 325–327, 1976.

[25] SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 21, n. 3, p. 660–674, 1991.

[26] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[27] BREIMAN, L. *Classification and regression trees*. [S.l.]: Routledge, 2017.

[28] WEBB, A. R.; COPSEY, K. D.; CAWLEY, G. *Statistical Pattern Recognition*. 3. ed. Hoboken, NJ: Wiley-Blackwell, 2011.

[29] BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565.

[30] PAL, M. Random forest classifier for remote sensing classification. *International journal of remote sensing*, Taylor & Francis, v. 26, n. 1, p. 217–222, 2005.

[31] LYONS, J. et al. *jameslyons/python_speech_features: release v0.6.1*. [S.l.]: Zenodo, 2020.

[32] CLOUGH, R. W.; PENZIEN, J. *Dynamics of Structures*. New York, NY: McGraw-Hill, 1975.

[33] BARANDAS, M. et al. Tsfel: Time series feature extraction library. *SoftwareX*, Elsevier, v. 11, p. 100456, 2020.

[34] HARRIS, C. R. et al. Array programming with numpy. *Nature*, Nature Publishing Group UK London, v. 585, n. 7825, p. 357–362, 2020.

[35] VIRTANEN, P. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020.

[36] BORATTO, T.; CURY, A.; GOLIATT, L. A fuzzy approach to drum cymbals classification. *IEEE Latin America Transactions*, v. 20, n. 9, p. 2172–2180, Jun. 2022.

[37] BORATTO, T. H. A. et al. Data-driven cymbal bronze alloy identification via evolutionary machine learning with automatic feature selection. *Journal of Intelligent Manufacturing*, Springer Science and Business Media LLC, nov 2022.