An Experimental Study on Applying Metamorphic Testing in Machine Learning Applications

Sebastião H. N. Santos
Instituto de Ciências Matemáticas e de Computação
São Carlos, SP
sebastiaohns@usp.br

Beatriz Nogueira Carvalho da Silveira
Instituto de Ciências Matemáticas e de Computação
São Carlos, SP
beatrizncs@usp.br

Stevão A. Andrade
Instituto de Ciências Matemáticas e de Computação
São Carlos, SP
stevao@icmc.usp.br

Márcio Delamaro
Instituto de Ciências Matemáticas e de Computação
São Carlos, SP
delamaro@icmc.usp.br

Simone R. S. Souza
Instituto de Ciências Matemáticas e de Computação
São Carlos, SP
srocio@icmc.usp.br

ABSTRACT

Machine learning techniques have been successfully employed in various areas and, in particular, for the development of healthcare applications, aiming to support in more effective and faster diagnostics (such as cancer diagnosis). However, machine learning models may present uncertainties and errors. Errors in the training process, classification, and evaluation can generate incorrect results and, consequently, to wrong clinical decisions, reducing the professionals' confidence in the use of such techniques. Similar to other application domains, the quality should be guaranteed to produce more reliable models capable of assisting health professionals in their daily activities. Metamorphic testing can be an interesting option to validate machine learning applications. Using this testing approach is possible to define relationships that define changes to be made in the application's input data to identify faults. This paper presents an experimental study to evaluate the effectiveness of metamorphic testing to validate machine learning applications. A Machine learning application to verify breast cancer diagnostic was developed, using an available dataset composed of 569 samples whose data were taken from breast cancer images, and used as the software under test, in which the metamorphic testing was applied. The results indicate that metamorphic testing can be an alternative to support the validation of machine learning applications.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**.

KEYWORDS

Metamorphic Test, Machine Learning, Experimental Study

ACM Reference Format:
Sebastião H. N. Santos, Beatriz Nogueira Carvalho da Silveira, Stevão A. Andrade, Márcio Delamaro, and Simone R. S. Souza. 2021. An Experimental Study on Applying Metamorphic Testing in Machine Learning Applications. In *Proceedings of SAST (SAST2020).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine Learning (ML) algorithms has become increasingly popular and used in a variety of applications due to the popularization of libraries that facilitated their use and the ability to automatically extract patterns from provided data. The great aptitude of these algorithms to make predictions from historical data, in addition to the high-speed and low-cost processing, allowed the advancement of several activity areas and the development of many important applications, such as cancer monitoring and treatment [12], identification of obstacles by autonomous vehicles [1] and even identification of feelings in speech [18].

Although ML applications show promising results, these techniques still have uncertainties that can reduce the motivation of the professionals to adopt in their daily use (bias, errors in the training process, classification and measurement of the model [9]) and can lead to wrong decisions. A specific characteristic of these applications is that they are non-deterministic, their results rely on the data provided, thus the constant addition of new data to improve models can impact in the output of the application. Furthermore, in this domain there are specifics properties, like robustness, result adequacy, data privacy, security and efficiency that need to be guaranteed in order to ensure reliability.

The use of software quality assurance processes is essential in the development of an application. In critical domains, such as autonomous vehicles and health treatments, ensuring that the application developed performs its purpose correctly is crucial to avoid unnecessary costs and even catastrophic losses [13]. These new contexts add challenges to the software testing activity, as it is necessary to identify the characteristics of these systems and their main types of faults, in order to allow the establishment of better software testing strategies.

The effectiveness of ML applications is often verified using a set of performance metrics based on the confusion matrix, which shows information about the real value and predicted rating done by a classification system. Examples of metrics are *accuracy*, which measures the rate of correct predictions under all predictions made, *precision*, which measures the probability that a positive prediction is actually positive, *recall*, which measures the probability of a positive sample was classified as positive and *F1 score*, which is a function of true-positive rate (precision) by positive predictive value (recall) [10].

During the software testing activity, the tester examines if the output produced by the software under testing is the output expected for the input data. In some situations, it is difficult to analyze the produced output and, therefore, it is hard to determine whether the program behaves as expected. This problem is known as "the oracle problem" [20]. An ML application has the characteristic to learn from the training data. Considering that new data are added through the time, the output of these applications might not be always the same. Therefore, an oracle built for an ML application maybe not reflect the actual result.

Metamorphic Testing is an existing technique to test software which does not possible to have an oracle. The technique consists of constructing test cases based on the input and output of an original test case [4]. The new test cases are produced making modifications to the input of the software. The relation of the original test case and the new one (a modified version from the original test case) are verified through a Metamorphic Relation (MR). If the output produced by a follow-up test case, generated based on a source test case, attend the constraints imposed by a given MR, we can conclude that the program behaved as expected. Otherwise, it is an indication of a possible fault in the software under testing [25].

Several works present case studies on the application of metamorphic testing to specific testing problems in different domains, such as web services and applications, computer graphics, simulation and modeling, and embedded systems [17]. The adoption of MT by ML applications is also growing and, to alleviate the oracle problem, many MRs have been proposed for particular problems, such as bio-genetics classifiers [5, 22], image classifiers [15, 23], and for deep learning applications [24].

This paper evaluates the efficiency of the metamorphic testing for the context of ML applications. Specifically, we are interested to answer the following research question: *"Is the use of metamorphic relations effective in detecting faults in a breast cancer machine learning classification model?"*. We implemented a set of MR proposed by Xie et al. [22] and evaluated its adequacy in an ML application used for breast cancer detection. The results allow us to discuss the experiences, benefits, and challenges of using this type of technique.

The rest of this paper is organized as follows: Section 2 presents related works. Section 3 provides background information about machine learning and metamorphic testing, including the definition of the metamorphic relations utilized in this work. Section 4 presents the details of the experiment, including the structure of the objectives, research question and hypothesis. Section 5 presents the instrumentation to develop the experiment. Section 6 presents the details of the execution of the experiment. Section 7 presents the results and discussion of the experiment. Section 8 presents the threats to the validity of the work and, finally, Section 9 brings some conclusions and presents future works.

2 RELATED WORKS

Several studies on the use of metamorphic testing in machine learning applications have been developed. These works mostly focus on defining modifications in the dataset that will generate test cases to verify the quality of the model under testing. Additionally, some works also present reports of the execution of the modified test cases to verify their applicability.

Chen et al. [5] investigated the use of MT in two applications of the bioinformatics domain. An unique MR was defined for each application and the goal was to analyze the behavior of the weight defined in edges, since the applications received a graph as input data. The MRs were applied to generate changes in the applications' entries and detect possible faults in both applications. The objective of the work was to present the facility to develop, implement and apply the MRs in the test activity.

Xie et al. [22] defined a set of 11 MRs based on the properties inherent to the *k* Nearest Neighbors and Naive Bayes algorithms. The objective of the work was to verify the existence of faults in machine learning algorithms, by conducting a case study using a fictitious dataset and the algorithms provided by the *Weka* tool. The case study revealed that the MRs revealed faults in the algorithms. To validate the effectiveness of the case study performed, the authors used mutation testing in the algorithms evaluated. The results of the work indicated that the use of MT together with cross-validation can decrease the possibility of faults in machine learning applications.

In the work of Dwarakanath et al. [8] were presented 8 MRs, 4 for each of the 2 applications used. The first application was an handwritten digits image classifier, developed using Support Vector Machine learning algorithm available in the scikit-learn library and the second application a classifier of images based on deep learning called Residual Network (ResNet). The authors provided mathematical proofs of all MRs and the case study used mutation testing to modify the dataset generated by the MRs, serving as a comparative in the analysis of the mutant programs. It was found that most of the mutant programs were identified through the use of MT.

We observe that most studies in the area intend to present new MRs, usually directed at a specific domain, and to validate the relationships through simple case studies. Thereby, it was identified a lack of studies carrying out well-defined experimental studies on the application of the MT in the machine learning domain. Therefore, in this work, we attempt to collaborate with the area, presenting a well-defined experimental study on the applicability of metamorphic relations, defined by Xie et al. [22], in a machine learning application in the health area.

3 BACKGROUND

In this section, we present the main concepts and elements related to machine learning used in this work, including algorithm used to develop the model used in the experiment (*K* Nearest Neighbors).

The definition of Metamorphic Testing is also presented including the details of the metamorphic relationships used in the experiment.

3.1 Machine Learning

Machine learning is defined as the automated process of extracting patterns from a given data set [11]. Machine learning algorithms that aim to classify an unknown value (output variable), using a set of known values (input variable) are called supervised classification algorithms [14]. To create a machine learning application first is necessary to create a model based on the data set available, this model is tested against samples of the dataset and then used in the application to predict unlabeled values.

The process to create a machine learning model classifier starts by selecting the target dataset D, which is populated by n samples. The dataset D is made up of x attributes (or features) $A = \{a_1, a_2, a_3, ..., a_x\}$ and m classes (or labels) $C = \{c_1, c_2, c_3, ..., c_m\}$, $m \geq 2$. Each sample is constituted by different attributes values and associated to an existent class in C. Then the dataset is divided into training set and test set, as the goal of the model is to classify the unlabeled data correctly the test set don't include the labels of the samples. The training set will be the input of the machine learning algorithm that will "learn" and make a model of the existing pattern in the data available. Then the test set will be the input of the model created, which will make predictions of the classes of each sample according to the values of its attributes.

The activity described above can also be used to evaluate the correctness of the predictions and the performance of the created model, some metrics are utilized here like the ones based on the confusion matrix and also the time taken by the model to make the prediction. A model that makes more right classifications than wrong and takes no substantial time in its activity is considered a good model to be used in application. The workflow of a machine learning classification can be visualized in the Figure 1.

The machine learning algorithm used in this work was the k Nearest Neighbors (kNN). In the kNN algorithm, given a training data set, is assumed that each sample has m attributes and n classes. The algorithm calculates the distance between each training sample and the test case, the Euclidean distance is generally used. The algorithm then selects the nearest k samples that are considered the nearest k neighbors. Then the proportion of each class of the nearest neighboring k is calculated and the class with the highest proportion is assigned as the test case class [14].

To asses whether the classification performed by the model was correct we used the balanced accuracy score. In a binary case, balanced accuracy is equal to the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate) [16]. The formula is presented above.

$$balanced\_accuracy\_score = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

3.2 Metamorphic Testing

Metamorphic Testing consists of modifying the input data of a given algorithm following the specification of constraints, in such a way that is possible to predict some characteristics of the new output, given the output of the original input. The metamorphic testing can be easily implemented in practice, by identifying a set of properties called metamorphic relations that defines configurations which relate various inputs and their resulting outputs for a target algorithm.

In a metamorphic testing a test case performed against a set of MRs and by sequence, test cases derived from the initial test case are performed in order to verify whether the restriction imposed by the MRs remains. If the restriction is violated, it is said that the program under test has a fault.

In the work of [22] eleven MRs were formally defined according to the behaviors of supervised machine learning classifiers. These MRs were used to execute the metamorphic testing presented in the experiment of this work. Each MR is defined below as they were in the original work:

- **MR - 0: Consistence with affine transformation:** To the values of any subset of attributes for each sample in the training data set S and the test case t_s apply an arbitrary affine transformation function, $f(x) = kx + b$, $(k/= 0)$.
- **MR - 1.1: Permutation of class labels:** If the original test case result is l_i, apply a permutation function to the result of the follow-up test case, the result of the follow-up case should be the permutation of l_i. The permutation function should perform one-to-one mapping between a class label in the set of labels L to another label in L.
- **MR - 1.2: Permutation of the attribute:** Permute the m attributes of the dataset of all the samples and the test data.
- **MR - 2.1: Addition of uninformative attributes:** For the test case t_s, suppose the result is $c_t = l_i$ for the test case. In the follow-up test case, add an uninformative attribute to each sample in S and respectively a new attribute in t_s. The output of the follow-up test case should still be l_i. An uninformative attribute is one that is equally associated with each class label.
- **MR - 2.2: Addition of informative attributes:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, add an informative attribute to each sample in S and t_s such that this attribute is strongly associated with class l_i and equally associated with all other classes. The output of the follow-up test case should still be l_i.
- **MR - 3.1: Consistence with re-prediction:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, append t_s and c_t to the end of S and C respectively. Resulting in the new training dataset S' and C', which will be the input of the follow-up case. The output of the follow-up test case should still be l_i.
- **MR - 3.2: Additional training sample:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, duplicate all samples in S with label l_i, as well as their associated labels in C. The output of the follow-up test case should still be l_i.
- **MR - 4.1: Addition of classes by duplicating samples:** For the test case t_s, suppose the result is $c_t = l_i$ In the follow-up input, duplicate all samples in S and C that do not have label l_i and concatenate an arbitrary symbol, for example "*", to the class labels of the duplicated samples. The output of the follow-up test case should still be l_i. Another derivative of this metamorphic relation is that duplicating all samples

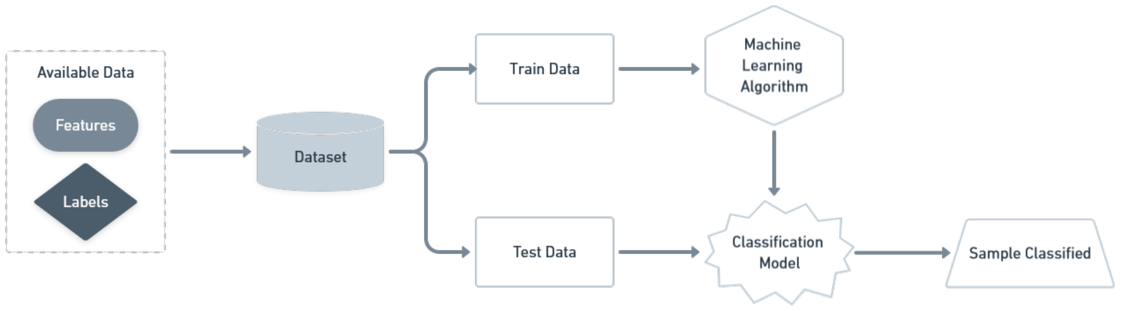

Figure 1: Machine Learning Classifier Application Workflow.

from any number of classes which do not have label l_i should not change the result of the output of the follow-up test case

- **MR - 4.2: Addition of classes by re-labeling samples:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, re-label some of the samples in S with labels other than l_i, through concatenating an arbitrary symbol, for example "*", to their associated class labels in C. The output of the follow-up test case should still be l_i.
- **MR - 5.1: Removal of classes:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, remove one entire class of samples in S of which the label is not l_i. The output of the follow-up test case should still be l_i.
- **MR - 5.2: Removal of samples:** For the test case t_s, suppose the result is $c_t = l_i$. In the follow-up input, remove part of some of the samples in S and C of which the label is not l_i. The output of the follow-up test case should still be l_i.

The metamorphic relations presented in the work of Xie et al. [22] were defined based on the characteristic of supervised machine learning classifiers. This paradigm is the main choice when dealing with datasets with categorical and numerical data (much of the data available). Also, the supervised machine learning classifier used was the same used in this work (kNN and also the Naive Bayes classifier). Those reasons, allied with the fact that the work is relevant (considering citation count and search results), were decisive to select the metamorphic relations defined in the work to use in this experiment.

4 EXPERIMENT DESIGN

In order to verify whether the metamorphic testing is applicable and efficient in guaranteeing the quality of machine learning applications, an experiment was conducted. A breast cancer diagnostic application was implemented using the kNN supervised classification model and a set of test cases were generated based on the MRs defined in the last section.

The experiment was conducted following the guidelines for experimental process proposed by Wohlin et al. [21]. Also, for structuring the experiment design and execution we applied the Goal / Question / Metric (GQM) model [3], which in its goal template divide the experiment in five parts: object of study, purpose, focus, perspective and context. The GQM template is described bellow:

- **Object of study**: assess the effectiveness of MRs in the implemented model.
- **Purpose**: evaluation of applicability of the MT in context of ML.
- **Focus**: effectiveness of the classification made by the model implemented.
- **Perspective**: academic.
- **Context**: researchers assessing the effectiveness of a machine learning model in a cloud-based execution environment (*Google Colaboratory*).

Specifically, the experiment was designed to investigate the following research question:

- **RQ:** *Is the use of metamorphic relations effective in detecting faults in a breast cancer machine learning classification model?*

In order to answer the research question, two variables were collected for this experiment, one dependent and one independent, as described below:

- **Dependent Variable**: the resulting classifications given by the ML model in a test case.
- **Independent Variable**: the dataset that will be modified in order to create the test cases based on the modifications defined by the MRs.

To enable the evaluation of the results, we have converted the research question of this experiment into the following hypotheses described below:

- **Null hypothesis**: The set of MRs evaluated is not effective for detecting faults in ML applications.
- **Alternative Hypothesis**: The set of MRs evaluated is effective for detecting faults in ML applications.

5 INSTRUMENTATION

As the goal assess the quality of an application in the health domain, we searched for a suitable dataset. Thus, the dataset *Breast Cancer Wisconsin (Diagnostic)* [7] was selected. This data set is composed by 569 samples and 30 attributes calculated from scanned images of breast mass which describes characteristics of the cell nuclei present in the images. The dataset has two classes and the distribution is given by 357 as benign and 212 as malignant.

This dataset was chosen for the experiment because its number of samples and attributes was found to be adequate to test the machine learning application in our context. The dataset contains numerical and categorical data which is ideal to be used in a supervised machine learning classifier model, which is the objective of the experiment. It allowed the training and testing sections of the model and the subsequent execution of the experiment not to extend over a long time, but also to be not too simple. Besides, it's a free well-known dataset used in the field of machine learning. Given that, was possible to create an ML model to classify samples with information about breast cancer mass is benign or malignant.

Several tools and libraries were used to develop the application evaluated in this experiment, one was *Google Colaboratory* or *Colab*, a free cloud service hosted by *Google*, that execute *Python* code and is focused in the development of artificial intelligence algorithms. The open-source machine learning library *Scikit.learn* [16] was also used, which has several machine learning algorithms implemented, such as the *k*NN that was used in the experiment, and additionally the quality assessment metrics used to evaluate the algorithms. Also, the library *Pandas* was used to manipulate and analyze the dataset [19]. These apparatus was chosen by its optimal capacity in building machine learning applications, it all gives a complete set of functionality to perform all the task necessary without being limited by hardware or manual implementation.

6 EXECUTION

As part of the experiment, a machine learning application to classify samples of breast cancer mass in benign of malignant was developed. The algorithm used to create the ML model was the *k*NN algorithm. The dataset used was imported in a *"notebook"* in the *Google Colab*. The components used for training and testing the model were implemented, as well as the functions to validate the correctness of the classification performed by the model, using the library *Scitkit.learn, Numpy* and *Pandas*. The dataset, implementation of the application, and the resulting data of the experiment can be accessed by the following link https://tinyurl.com/tmam2020. The Figure 2 describes an illustration of the workflow used in the development of the experiment.

We used the leave-one-out cross-validation technique [2] (to assess the generalization of a model, based on the data set) in the training and testing process of the ML application. This cross-validation technique separates the data into two parts, one for training the model and the other for testing, in the following configuration: considering that the dataset has n samples, the part for training the model will be equal to $n - 1$ and the remaining sample will be used as a test, this procedure is repeated n times, excluding at each moment a different observation.

The configuration of the different test cases (the original test case and the 11 test cases based on the modifications defined by the metamorphic relation) were implemented making the necessary changes in the application's characteristics. The modifications in the dataset defined by each of the metamorphic relations was implemented following the characteristics specified bellow:

- MR – 0: Consistence with affine transformation, was applied the following transformation function to all samples in the dataset, $f(x) = 2x + 2$.
- MR - 1.1: Permutation of class labels, was defined a permutation function, which when the class label of the sample was defined as benign it will permute to malignant and when the class label was malignant it will permute to benign.
- MR - 1.2: Permutation of the attribute, the attributes was permuted following their position in the dataset, if an attribute was in the p position it was permuted with the attribute in the $p - 31$ position.
- MR - 2.1: Addition of uninformative attributes, was added to all samples an uninformative attribute named *"uninformative"* with the value of a random number equal in all samples.
- MR - 2.2: Addition of informative attribute: was added to all samples an informative attribute named *"informative"* which for the samples with class label defined as benign the value of the attribute was 0 and for the samples with class label defined as malignant the value of the attribute was 1.
- MR - 3.1: Consistence with re-prediction: when the sample s was to be classified by the model, the same x sample already classified in the original test case was re-added to the dataset.
- MR - 3.2: Additional training sample: if the sample s is classified as benign, all samples classified as benign was duplicated. And the contrary, if the sample s is classified as malignant, all samples classified as malignant was duplicated.
- MR - 4.1: Addition of classes by duplicating samples: if the sample s is classified as benign, all samples classified as malignant was duplicated, the duplicated samples was relabeled to malignant*. And the contrary, if the sample s is classified as malignant, all samples classified as benign was duplicated, the duplicated samples was relabeled to benign*.
- MR - 4.2: Addition of classes by re-labeling samples: if the sample s is classified as benign, half of the samples classified as malignant was relabeled to malignant*. And the contrary, if the sample s is classified as malignant, half of the samples classified as benign was relabeled to benign*.
- MR - 5.1: Removal of classes: if the sample s is classified as benign, all samples classified as malignant was removed from the dataset. And the contrary, if the sample s is classified as malignant, all samples classified as benign was removed from the dataset.
- MR - 5.2: Removal of samples: if the sample s is classified as benign, half of the samples classified as malignant was removed from the dataset. And the contrary, if the sample s is classified as malignant, half of the samples classified as benign was removed from the dataset.

The original test case consisted of the input of the original dataset (without modifications) in the machine learning process. Including the cross-validation, which resulted in 569 execution of the machine learning workflow, which, in each time, 568 different samples were used for training the model and 1 different sample was used to test the model. For each metamorphic relation was defined as one test case, including its modified dataset. Thus, considering the dataset used in the developed application, 569 test observations were made for each of the 12 configurations of the test cases (the original test case and the 11 MR test cases), resulting in 6828 observations, which, in the machine learning workflow, the process was followed as the same in the original test case. As a result of running the 12 test cases, two lists were returned for each test case.

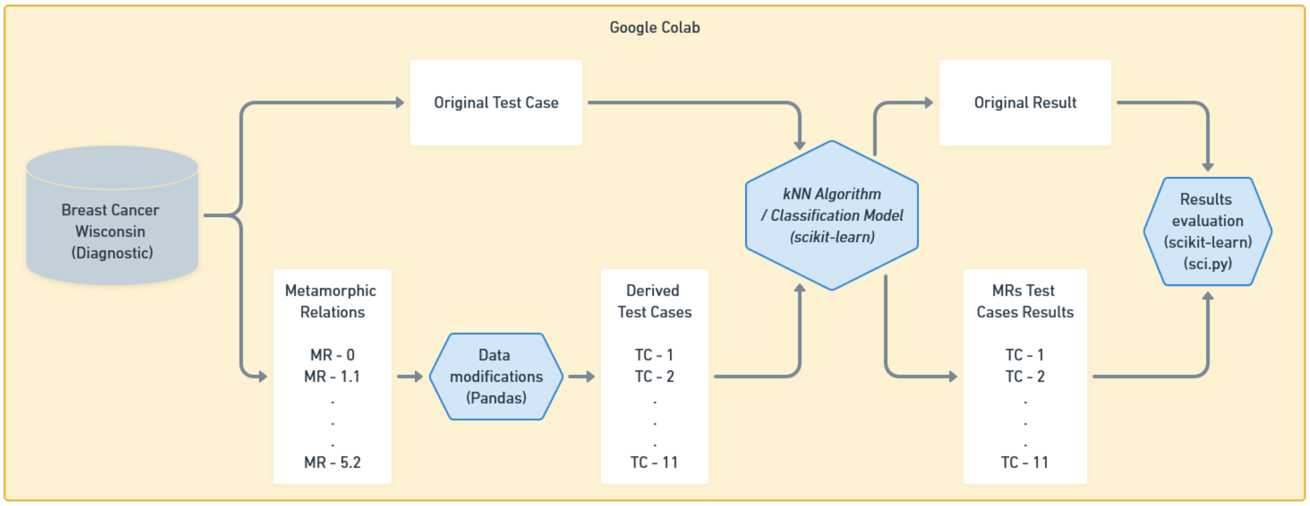

Figure 2: Development workflow of the experiment.

One list contains each of the classifications made for each test case observation and the other list contains the model accuracy for each test case observation. Accuracy is the metric that assesses whether the classification performed by the model was correct by comparing the model classification result with the true classification.

7 RESULTS AND DISCUSSION

As previously reported, the class distribution of the samples in the dataset is given by 357 samples as benign and 212 as malignant. When running the model in the original test case 419 samples were classified as benign and 150 as malignant. This result of the classification model gives us an accuracy score of 78,56%, which means that the model created classified this percentage of the samples correctly according to their real specified values in the original dataset. The results are presented in Table 1.

Table 1: Results of the original test case

Benign Classifications	Malignant Classifications	Model Accuracy
419	150	78,56%

For each one of the 11 selected metamorphic relations was generated one test case, where the original dataset was modified, and the 569 samples of the modified dataset were used as the input of the classification model. In Table 2 the results of the test cases are presented, showing the number of the samples classified as benign and malignant and the accuracy score of the model with the modified dataset as input. The "results similarity" metric shows the percentage of the similarity of the results of the original test case and the results of the test case generated by the metamorphic relation. This metric measures the impact of the modifications specified in the metamorphic relations on the results of the model.

Table 2: Results of MR-based test cases

MR	Benign Classifications	Malignant Classifications	Model Accuracy	Results Similarity
0	419	150	78,56%	100%
1.1	419	150	78,56%	100%
1.2	419	150	78,56%	100%
2.1	419	150	78,56%	100%
2.2	419	150	78,56%	100%
3.1	419	150	78,56%	100%
3.2	419	150	78,56%	100%
4.1	414	155	76,45%	85,41%
4.2	419	150	78,56%	100%
5.1	419	150	78,56%	100%
5.2	401	168	85,24%	93,32%

We used Friedman's statistical test to detect whether or not there is a difference between the original results to the results of the modified test cases. Friedman's statistical test is a non-parametric statistical procedure to compare more than two related samples. Significant results may lead to the conclusion that at least one of the samples is different from the other samples [6]. The accuracy score of all test cases were used in the Friedman's test together, the results of the Friedman's test are presented in Table 3, which resulted in a p-value smaller than the 0.05 significance level.

Table 3: Friedman's test results

statistic	p-value	alpha
318.895	0.000	0.05

As Friedman's test p-value was less significant than the alpha value, it means that at least one of the test cases has results that are different from the others. If the discrepancy is obtained concerning the original test case, this result could indicate the presence of faults.

Friedman's test does not indicate which of the modified test cases has a different result compared to the original test case. Therefore, it was necessary to apply another method to identify the distinct differences between the original test case and each one of the modified test cases based on the metamorphic relations. In this way, the Wilcoxon signed ranks test was used to verify the distribution similarity of the classification of the original test case compared to the modified test cases. This statistical test is a non-parametric statistical procedure to compare two samples that are related [6]. The results of the classification model with the two different inputs, the original dataset and a modified dataset were analysed. The results are presented in Table 4, where the alpha value used was equal to 0.05.

Table 4: Results of the Wilcoxon test on MR-based test cases

Metamorphic Relation	statistic	p-value
0	81082.500	1.000
1.1	81082.500	1.000
1.2	81082.500	1.000
2.1	81082.500	1.000
2.2	81082.500	1.000
3.1	81082.500	1.000
3.2	81082.500	1.000
4.1	77725.500	0.331
4.2	81082.500	1.000
5.1	81082.500	1.000
5.2	70623.000	0.003

Most of the modified test cases generated by the metamorphic relations obtained a p-value greater than the alpha value and equal to 1 in the Wilcoxon test, proving that their resulting classifications are identical to the classification of the original test case. On the other hand, the p-value scored in the Wilcoxon test in the comparison of the original test case with the modified test cases based on the metamorphic relations 4.1 and 5.2 was different to 1, in the case of the metamorphic relation 5.2 the p-value was significantly different as it was lower than the alpha value. As defined by all the metamorphic relations, the results of the modified test cases should be the same as the original test case. As the modified test case based on the metamorphic relations 5.2 was significantly different from the original test case, this result may indicate that there is a fault in the model created.

Friedman's test identified a real difference among the test cases. The Wilcoxon test pointed out a significant difference between the original test case and the modified test case based on the metamorphic relation 5.2. These results may indicate the presence of a fault in the model created and, therefore, we can reject the null hypothesis. Thus, in this experiment, we have evidence about the alternative hypothesis indicating that the set of metamorphic relations evaluated in the experiment is capable of detecting faults in a machine learning model.

Considering that the metamorphic relation 5.2 defines the removal of part of some of the samples in the dataset for the training phase, this specific modification may detect a fault in the model created. The reduction of the samples in the training phase caused an impact in the classification model, as it had fewer samples to learn from the data pattern. The metamorphic relation 4.1, which also defines changes in the number of the samples in the dataset (add samples), caused a deviation in the results classification of the model. Given these two observations, we found that the fault that was found may be in the power of generalization of the algorithm, in this case, when the number of samples available for the algorithm changes, the classification realized by the model is affected.

These results indicate that the algorithm is sensitive to the size of the dataset. However, it may not mean that more or less samples can improve the classification realized by the model. As in the test case derived from the metamorphic 4.1, which increase the number of samples, the accuracy score was almost the same as the original test case, and in the test case derive from the metamorphic relation 5.2, which has fewer samples to train the model, the accuracy score was greater than the original test case. The accuracy score of test case based on the MR 5.2 could be explained by a behavior called over-fitting, which means that the model fits very well with the previously observed data, resulting in a better classification for the dataset provided but may not classify correctly data from other sources.

In general, the metamorphic testing was considered a useful approach to test the effectiveness of the machine learning classification model. The metamorphic relations selected defined a great set of modifications to be applied in the dataset number of attributes, class labels, and samples. It was possible to detect the effectiveness of the model by observing the lack of deviation on the classification results when the data used in the training process was changed, a sensible activity in a machine learning application. And a singular characteristic of the model evaluated indicated as a fault, that its sensitiveness to the sample numbers in the training phase.

8 THREATS TO VALIDITY

This work is subject to the threats to the validity, thus the results must be interpreted carefully. Bellow we discuss the validity concerns based on classification proposed by Wohlin et al. [21].

- **Construction Validity**: to conduct the experiment, it was necessary to perform some changes in the MRs originally proposed in [22]. However, the purpose and basic structure of the MRs was maintained. Besides, all the technologies and metrics used to consolidate the results of the experiments (open source libraries like *Scikit Learn*, *Numpy*, *Pandas* and *Scipy*) constitute the current state of practice for the development and evaluation of ML algorithms for *Python* programming language. Despite the use of different set of metrics, to evaluate the machine learning model, could alleviate the misgiving of the results, the metric used is still a great measurement of the classification results as its composed by other metrics and considers general characteristics of the model.
- **Internal Validity**: to mitigate a possible influence on the variables of the experiment we chose to use a third-party

well-known open-source dataset, that has already been successfully used in other experiments. This reduces the possibility that the results presented in the experiment may have suffered some type of influence due to the data, since, as described previously (section 3), the success of ML models is directly related to the dataset used in its creation.

- **External Validity**: a common problem faced in experiments in software engineering context is related to the generalization of the results presented by the studies. Despite using a dataset widely known in the literature, the dataset chosen for the development of the model training and the creation of the test cases consists exclusively of breast mass images data and specifically deals with breast cancer detection in the health area. Thus, it is possible that our results may not be generalized to types of applications and datasets built for other domains.
- **Conclusion Validity**: concern the relationship between the treatment and the outcome obtained in the experiment. To mitigate possibles threat in the conclusion of the experiment, we designed the experiment to collect data from dependent and independent variables, in addition to performing statistical tests that provide evidence regarding the generalization of results.

9 CONCLUSION

This work presented the design, planning and execution process to validate the use of a MT approach for ML applications. This specific type of applications is part of the group of software domains that present complex outputs - that is, they are difficult to identify and interpret, causing the test oracle problem. Therefore, we evaluated the use of MRs, previously used in other studies, as a way to verify the effectiveness of the application of metamorphic testing in the context of machine learning applications.

The major goal of the experiment was to detect the applicability of metamorphic relations by creating test cases based on the selected metamorphic relations for a machine learning model of a specific domain, classification of breast cancer as malignant or benign. The results of the experiment showed that the results of the test cases derived from the metamorphic relations were similar to the results of the original test case in most cases. Differences in the results were detected only in two test cases generated by specific metamorphic relationships.

We performed Friedman's statistical test to verify if it was possible to identify differences between the original dataset and those generated based on the evaluated MRs. After observe the difference between the groups, we applied the Wilcoxon test individually, between the original dataset and those generated from the MRs. The results lead us to refute the null hypothesis of the experiment and therefore, point to some evidence regarding the alternative hypothesis, which states that the set of MRs evaluated is capable of detecting faults in ML applications.

From the results found in the experiment and in the statistical tests used, the authors considers that the metamorphic relations evaluated are of great value and can be used for assist the software testing activity process for machine learning applications. Especially due to the characteristics of this type of application on constantly perform modifications to the dataset used for training and testing of the machine learning model, as these modifications can have a great impact on the models and evoke new faults.

Despite the evidence presented, the experiment must be replicated in different domains and implementations of machine learning applications so that the results obtained are verified and generalized. As a future work, the author hopes to replicate the experiment in other different domains and implementations, as well as adding more metamorphic relationships, for a more complete verification of the models. We also intent to apply the metamorphic testing to verify the integrity of the metamorphic relations even when there's no actual impact on the classification of the ML model.

ACKNOWLEDGMENTS

Sebastião H. N. Santos and Beatriz Nogueira Carvalho da Silveira was funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). Stevão A. Andrade research was funded by Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP), process number 2017/19492-1. The authors also thank the support of FAPESP process number 2019/06937-0 and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) process number 308615/2018-2 and 312922/2018-3. Also, acknowledge Ellen Francine Barbosa for the instructions about experimental study.

REFERENCES

[1] Marco Allodi, Alberto Broggi, Domenico Giaquinto, Marco Patander, and Antonio Prioletti. 2016. Machine learning in tracking associations with stereo vision and lidar observations for an autonomous vehicle. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, IEEE, Gotenburg, Sweden, 648–653.
[2] Sylvain Arlot, Alain Celisse, et al. 2010. A survey of cross-validation procedures for model selection. *Statistics surveys* 4 (2010), 40–79.
[3] Victor R Basili1 Gianluigi Caldiera and H Dieter Rombach. 1994. The goal question metric approach. *Encyclopedia of software engineering* (1994), 528–532.
[4] T. Y. Chen, S. C. Cheung, and S. M. Yiu. 1998. Metamorphic Testing: A New Approach for Generating Next Test Cases. arXiv:2002.12543 [cs.SE]
[5] Tsong Yueh Chen, Joshua WK Ho, Huai Liu, and Xiaoyuan Xie. 2009. An innovative approach for testing bioinformatics programs using metamorphic testing. *BMC bioinformatics* 10, 1 (2009), 24.
[6] Gregory W Corder and Dale I Foreman. 2011. Nonparametric statistics for non-statisticians.
[7] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[8] Anurag Dwarakanath, Manish Ahuja, Samarth Sikand, Raghotham M Rao, RP Jagadeesh Chandra Bose, Neville Dubash, and Sanjay Podder. 2018. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*. ACM, Amsterdam, Netherlands, 118–128.
[9] Milena A Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. 2018. Potential biases in machine learning algorithms using electronic health record data. *JAMA internal medicine* 178, 11 (2018), 1544–1547.
[10] Guy S Handelman, Hong Kuan Kok, Ronil V Chandra, Amir H Razavi, Shiwei Huang, Mark Brooks, Michael J Lee, and Hamed Asadi. 2019. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology* 212, 1 (2019), 38–43.
[11] John D Kelleher, Brian Mac Namee, and Aoife D'arcy. 2015. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* MIT press, Cambridge, United Kingdom.
[12] Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis, and Dimitrios I. Fotiadis. 2015. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* 13 (2015), 8 – 17. https://doi.org/10.1016/j.csbj.2014.11.005
[13] Patrick McDaniel, Nicolas Papernot, and Z Berkay Celik. 2016. Machine learning in adversarial settings. *IEEE Security & Privacy* 14, 3 (2016), 68–72.
[14] Iqbal Muhammad and Zhu Yan. 2015. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing* 5, 3 (2015), 946–952.
[15] S. Nakajima and H. Bui. 2016. Dataset Coverage for Testing Machine Learning Computer Programs. In *2016 23rd Asia-Pacific Software Engineering Conference*

(APSEC). IEEE Computer Society, Los Alamitos, CA, USA, 297–304. https://doi.org/10.1109/APSEC.2016.049

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[17] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. 2016. A survey on metamorphic testing. *IEEE Transactions on software engineering* 42, 9 (2016), 805–824.

[18] Mohammad Shami and Werner Verhelst. 2007. An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech. *Speech Communication* 49, 3 (2007), 201 – 212. https://doi.org/10.1016/j.specom.2007.01.006

[19] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). Austin, Texas, US, 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a

[20] Elaine J Weyuker. 1982. On testing non-testable programs. *Comput. J.* 25, 4 (1982), 465–470.

[21] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media, Berlim, Germany.

[22] Xiaoyuan Xie, Joshua WK Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. 2011. Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software* 84, 4 (2011), 544–558.

[23] Xiaoyuan Xie, Zhiyi Zhang, Tsong Yueh Chen, Yang Liu, Pak-Lok Poon, and Baowen Xu. 2020. METTLE: a METamorphic testing approach to assessing and validating unsupervised machine LEarning systems. *IEEE Transactions on Reliability* (2020), 1–30.

[24] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE 2018)*. Association for Computing Machinery, New York, NY, USA, 132–142. https://doi.org/10.1145/3238147.3238187

[25] Zhi Quan Zhou, DH Huang, TH Tse, Zongyuan Yang, Haitao Huang, and TY Chen. 2004. Metamorphic testing and its applications. In *Proceedings of the 8th International Symposium on Future Software Technology (ISFST 2004)*. Software Engineers Association Xian, China, Citeseerx, Xian, China, 346–351.