International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

The Adequacy Assessment of Test Sets in Machine Learning using Mutation Testing

Hoijin Yoon

Abstract*: The accuracy is computed by applying the test dataset to the model that has been trained using the training dataset. Thus, The test dataset in machine learning is expected to be able to validate whether a trained model is sufficiently accurate for use. This study addresses this issue in the form of the research question, "how adequate is the test dataset used in machine learning models to validate the models." To answer this question, the study takes seven most-popular datasets registered in the UCI machine learning data repository, and applies the data sets to the six difference machine learning models. We do an empirical study to analyze how adequate the test sets are, which are used in validating machine learning models. The testing adequacy for each model and each data set is analyzed by mutation analysis technique.

Keywords* : Software testing, Mutation analysis, Machine learning, Test adequacy.

I. INTRODUCTION

Several recent studies have applied software testing methods to machine learning (ML) testing. *DeepXplore* was introduced to test deep neural networks (DNNs), and it generates more effective test cases by utilizing its new neuron coverage [1]. *DeepMutation* [2] employs mutation testing and generates mutation operators for deep learning systems at both the source and model levels. While these studies focus on testing ML models by using traditional testing approaches, this study aims to determine how adequate the testing set is for developing ML models.

ML develops a "model" instead of an actual program in the traditional software development sense. The model is built via ML and the process is largely divided into two steps: (1) model completion through data training and (2) testing to validate the accuracy of the completed model [3]. In software testing, test cases are designed from the software test requirements. Thus, the efficiency of the testing is dependent on which test cases are designed and used. A well-designed test case must trigger a fault in the software under test (SUT), causing a failure. An adequacy assessment is necessary to evaluate the quality of the test cases designed in this manner. That is, among the current set of test cases, the number of cases that can detect software faults must be evaluated for adequacy [4].

However, the test set used in the second step of the ML model building process is built via simple quantitative division without any test design [5]. In the initial stage, a portion of the dataset is allocated and defined as test data, which are not used for training. The remaining data are used to train the ML models. The two parts of the dataset are called the training set and the test set. The feature values of the test set are input into the model, and then it is determined whether the results predicted by the model are equal to the corresponding label values of the test set. This is referred to as accuracy.

If the accuracy is above a predetermined level, the model is considered to be well built; otherwise, it is considered to be unusable and should be rebuilt. Therefore, the test set should be sufficiently adequate to distinguish between correct and incorrect models.

In this study, the adequacy of the test sets in ML was evaluated by mutation analysis, which is a typical technique for such cases [6,7,8]. It examines how well a test case distinguishes a mutant from the original program. The adequacy was measured in an experiment and the final 1260 adequacy values were analyzed.

II. EXPERIMENT

A. Datasets

The well-established University of California Irvine (UCI) ML Repository (https://archive.ics.uci.edu/ml/) is a collection of databases, domain theories, and data generators that are used for the empirical analysis of ML algorithms. The archive has been cited over 1000 times, making it one of the top 100 most cited papers in computer science [9]. In the repository, "Most Popular Data Sets" are ranked based on the number of downloads since 2007. The top seven datasets in this repository were selected for this experiment. The selected datasets have been used in many studies [10,11,12]. As described in Table-I, the seven selected experimental datasets are of different sizes from Iris (data volume: 150) to Adult (data volume: 48,842). Additionally, the feature analysis complexity of each dataset varies from simple Iris with 4 features to Breast Cancer Wisconsin with 32 features. Six typical models were built using these datasets: support vector machine (SVM), Gaussian Naïve Bayes, *latent Dirichlet allocation* (LDA), K-nearest neighbors (KNN), decision tree, and logistic regression.

B. Mutation Operators

Unlike in source code mutation, it was necessary to mutate data in this experiment. If numbers or characters are changed to incorrect values, they are likely to become meaningless dirty data.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Hoijin Yoon*, Department of Computer Engineering, Hyupsung University, Hwaseung, Kyunggi, South Korea. Email: hjyoon@uhs.ac.kr

Therefore, the feature values of the data were left intact and a training set for mutants was created by applying a shuffle operator that replaces labels with values within the range of label value. In this case, the training set created has shuffled labels and the model built via learning with this training set becomes a mutant model. Original and mutant models are required for mutation analysis. The resulting values for the two models should be different from each other as the respective training sets are different. Therefore, the test data are said to perform well when the two models can be differentiated. That is, when the predicted values generated from the models are different, the data are deemed adequate [4].

C. Models

The original model (denoted as M_0) was the model built with the training set of original data to which the mutation operation was not applied. The mutant model (denoted M_1~M_n) was built with the shuffled training set created by applying the mutation operation to the original data. It was used to calculate the mutation scores by creating n mutants. Thirty mutants (n=30) were created for a single M_0 for the experiment. The 30 different mutated datasets were created by shuffling the original data used in building M_0. In the experiment, there were 42 M_0s for the seven datasets and six models and M_1~M_{30} for each M_0. Thus, 1,260 models (*30 mutants * 7 datasets * 6 models*) were used in the experiment, where the mutant model is denoted as (M_i, where i = {1…n}).

D. Implementation

The pseudocode of the program used to evaluate the adequacy through mutation analysis is presented below. The program was implemented in Python (ver. 3.6) in the Geany® environment (IDE) [13] and using the scikit-learn library [14] for the ML models. The values were stored in a .csv file for each model and charts were drawn using the matplotlib library [15].

```
For p in [1..7 : 7 sets of data from ML repository, D1…D7]
1 Construct Tp (Training set selected from Dp) and Vp
  (Validation set in Dp) from Dp.
2 Build M0 learning Tp.
3 For M0 in [6 types of models]
  3.1 For q in [1..30]
    3.1.1 Create training set Tq by applying shuffle operation to
          Tp.
    3.1.2 Build Mq that has learned Tq.
    3.1.3 If the predictions of M0 and Mq are different after
          applying Vp, Mq is killed. Evaluate how many of the
          data belonging to Vp are killed as adequacy.
```

III. ADEQUACY ASSESSMENT

The adequacy values in this experiment were calculated using the following equation [4].

$$Adequacy(i) = \frac{|\{x|\ M_i(x) \neq\ M_0(x)\}|}{\text{The number of data in the test set}} \times 100$$

The equation was applied to perform adequacy assessments to determine how well the test set distinguished between M_0 and each mutant ($M_{1..30}$). For example, a decision tree model, M_0, was built which was trained with 80% of the Iris data and adequacy was evaluated for the remaining 20% of the test set. Then, a mutant model, M_1, was built which was trained with 80% of the Iris data that had been modified through the mutation operation. Among the data in the test set used in the evaluation, the number of data points that differed M_1 were examined. The percentage of the test set differed between M_0 and M_1 was measured and recorded as the adequacy of M_1. This process was repeated for M_2 up to M_{30}; hence, the adequacy of the test set of the Iris data was measured 30 times in the decision tree model.

A. Adequacy by Dataset

The adequacy was measured for a particular model trained with a particular dataset. The measured adequacy values for the 1,260 models are represented as boxplots in Fig. 1. The x-axis represents the dataset used and the number in parentheses after each dataset name indicates the data size. The datasets are listed in order of increasing size.

As seen in Fig. 1, the adequacy deviation for one validation set is very large. For the Iris dataset in the decision tree model, the adequacy of the validation set varies greatly depending on the mutants, $M_{1...30}$, i.e., from a maximum of 76.67% to a minimum of 46.67%. In particular, the Gaussian Naïve Bayes model shows large deviations. The Car Evaluation and Adult datasets show consistently small deviations and adequacy. The adequacy values are below 50%, which indicates that less than half of the data in the testing sets for the Car Evaluation and Adult datasets can distinguish between the trained models and their mutants.

B. Adequacy by Mutant Features

For the cases with a relatively high adequacy in terms of the mean value, there were large deviations in the adequacy values for the 30 models. Adequacy line plots were drawn for each mutant to confirm whether the adequacy was significantly affected by the mutant features. The confirmation was attempted to determine whether the adequacy of a particular mutant was consistent with each dataset; however, as seen in the chart, the patterns of high and low adequacy continued regardless of the mutant. For a more in-depth investigation, the 30 adequacy values of each dataset were listed for the decision tree model and the corresponding mutants are displayed in Fig. 2. The x-axis represents the 30 mutants, $M_{1..30}$, and the y-axis represents the adequacy values for each mutant. As shown in Fig. 2, the pattern in the adequacy values does not appear to be consistent for the mutants. For example, in the logistic regression model, the lowest adequacy of the Iris dataset is obtained for M_{17}; however, the adequacy for the Wine dataset is high for M_{17}. Thus, the adequacy values and mutant features are irrelevant.

C. Adequacy by Model

Fig. 3 shows boxplots of the adequacy of each model in the same dataset. As seen, the mean adequacy values are constant for the Iris and Wine datasets, which are relatively small, with sizes of 150 and 178, respectively. However, in cases where the mean values of adequacy are not constant, certain mean values are displayed only when the data size is relatively large. Variations in the adequacy for each model are observed for the Adult and Car Evaluation datasets, both of which have large data sizes.

However, for both large and small deviations, the mean values are all below 70%. In particular, they are below 40% in all models for the Adult dataset.

D. Threats of Validity

The validation size, which refers to the portion of the dataset allocated to the testing set from the entire dataset, must be set for each experiment. If the validation size is 0.2, 80% of the entire dataset is used as the training set and 20% is used as the testing set. As the adequacy of the testing set was evaluated in this study, the experimental results can be affected by the validation size used. Therefore, we conducted a preliminary experiment to identify any differences in the accuracy depending on the validation size. The results are shown in Fig. 4. The accuracy values measured for validation sizes of 0.1, 0.2, 0.3, or 0.4 are represented by boxplots for each model and dataset. The values are found to be constant with little variation. This means that the validation size may not have a significant effect on the outcome. Therefore, the validation size in the experiment was arbitrarily set as 0.2.

In addition, a mutant originates from a single mutation operator. It can be explained by the coupling effect, which is the theory that a simple mutant generated by a single mutation operator can cover complex mutants applied simultaneously by multiple operators [16].

For external validation, popular datasets were used in this experiment and very common classification models were built. The functions provided by scikit-learn were utilized and general situations were reproduced without setting any optional values.

IV. CONCLUSIONS

ML is used to create a model, train the model with a training set, and then test the model with a testing set. The training set and testing set are generated by dividing the original dataset into two parts. In this study, 80% of the original dataset was used as the training set and the remaining 20% was used as the testing set. The adequacy of the testing set was evaluated via mutation analysis. The average adequacy values obtained from the experiments are summarized in Table-II. The adequacy was below 50%, which means that only less than 70% of the data in the test set can distinguish between the original model, M_0, and the mutant, M_i. An adequacy of less than 50% is considered insufficient for effective testing.

In order to identify whether there are any consistent rules for adequacy, dataset size, and model type prior to the above mean analysis, boxplots and line plots from various angles were drawn and analyzed. The analysis results showed that the adequacy of the testing sets varied at an unpredictable level rather than being consistent with the rules according to model type and data size. Therefore, considering the adequacy of the test sets, test set configuration through current quantitative division techniques should be avoided. Further, there is a need to qualitatively select the test set and develop criteria for selecting test cases in order to guarantee a certain level of adequacy.

ACKNOWLEDGMENT

This work was supported by the Hyupsung University Research Grant of 2019 (2019-0041).

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03034557).

REFERENCES

1. K. Pei, Y. Cao, J. Yang, and S. Jana, “DeepXplore: Automated whitebox testing of deep learning systems,” *Proc. ACM SOSP*, pp.1-18, 2018.
2. L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, J. Liu, J. Zhao, and Y. Wang, “DeepMutation: Mutation testing of deep learning systems,” *Proc. ISSRE*, 2018.
3. A. Muller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, Beijing, 2016
4. A. P. Mathur and W. E. Wong, “An empirical comparison of data flow and mutation-based test adequacy criteria,” *Software Test Verification Reliability*, vol.4, pp. 9-31, 1994
5. L. P. Coelho and W. Richert, *Building Machine Learning Systems with Python*, Packt publishing, Birmingham, 2013.
6. R. A. DeMillo, R. Lipton, and F. Sayward, “Hints on test data selection," *Computer*, vol.11, no.4, pp. 34-41, 1978.
7. R. A. DeMillo, R. Lipton, and F. Sayward, “Program Mutation: A New Approach to Program Testing,” *Infotech State of the Art Report, Software Testing 2*, p.107-126, 1979.
8. T. A. Budd, “Mutation analysis of program test data,” Ph.D. thesis, Yale University, New Haven, Connecticut, 1980.
9. D. Dua and E. Karra Taniskidou, UCI Machine Learning Repository,
10. R. A. Fisher, “The use of multiple measurements in taxonomic problems," *Ann. Hum. Genet.*, vol.7, no.2, pp.179-188, 1936
11. R. O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley Sons, New York, 1973.
12. B. V. Dasarathy, “Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-2, no.1, pp.67-71, 1980.
13. Geany, https://www.geany.org/
14. Scikit-learn Library, https://scikit-learn.org/stable/
15. Matplotlib library, https://matplotlib.org/
16. A. J. Offutt, “The coupling effect: Fact or fiction," *ACM SIGSOFT Software Engineering Notes*, vol.14, no.8, pp.131-140, 1989.

AUTHORS PROFILE

Hoijin Yoon She is currently an associate professor in the Department of Computer Science and Engineering at Hyupsung University. She received the B.S. degree in Computer Science from Ewha Womans University and the M.S. and Ph.D. degrees in Computer Science and Engineering from Ewha in Korea. Her research interests are in software engineering with particular emphasis on testing. Her current research project is on V&V in Interactions between Intelligent things and Human.

Table-I : The seven most-popular datasets in the UCI machine learning repository

Subject	Size	# of features	Predicted attribute	Description
D1: Iris	150	4	class of iris plant.	The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
D2: Adult	48842	14	whether a person makes	Extraction was done by Barry Becker from the 1994 Census database.
D3: Wine	178	13	Type of wine (3types)	The analysis determined the quantities of 13 constituents found in each of the three types of wines.
D4: Car Evaluation	1728	6	Acceptability (unacc, acc, good, vgood)	a simple hierarchical decision model : Expert system for decision making.
D5: Breast Cancer Wisconsin	569	32	Diagnosis (M = malignant, B = benign)	Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.
D6: Wine Quality	4898	12	Score for Wine Quality	Various test data of red or white wine are set as features and the wine quality is scored up to 10 points."
D7: Heart Disease	303	13	diagnosis of heart disease	Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

Table-II : Mean Adequacy

Model \ Subject	Iris	Wine	Heart Disease	Breast Cancer	Car Evaluation	Wine Quality	Adult	Average
DecisionTree	66.44	66.29	65.72	47.57	48.2	65.19	37.27	56.67
KNN	66.22	63.33	36.5	39.38	53.68	52.28	20.47	47.41
LDA	65.88	63.79	47.05	34.97	35.26	56.16	19	46.02
Logistic Regression	62.11	65.18	32	39.61	26.01	51.69	8.15	40.68
Gaussian Naïve Bayes	65.33	61.66	61.55	48.01	45.92	62.11	11.25	50.83
SVM	75.88	62.12	36.16	29.29	32.36	53.78		48.27
Average	66.98	63.73	46.50	39.81	40.24	56.87	19.23	48.31

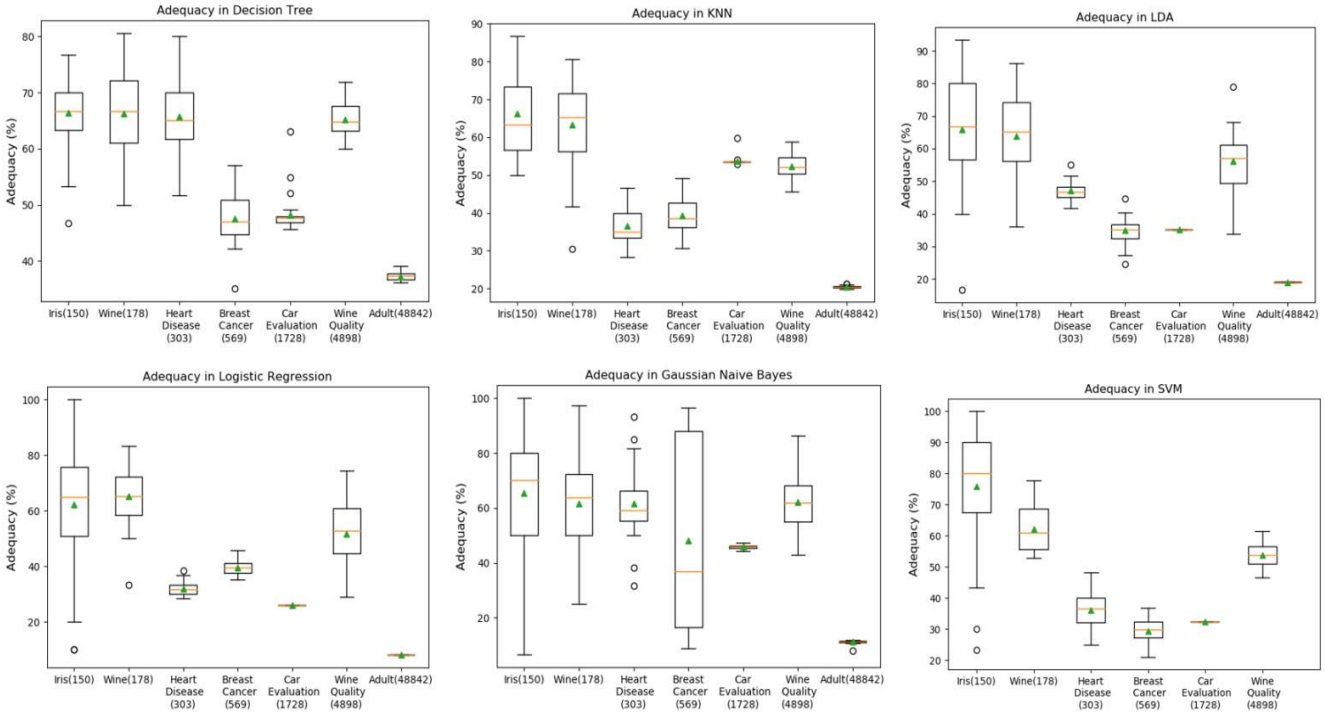

Fig. 1. Distribution of adequacy in models

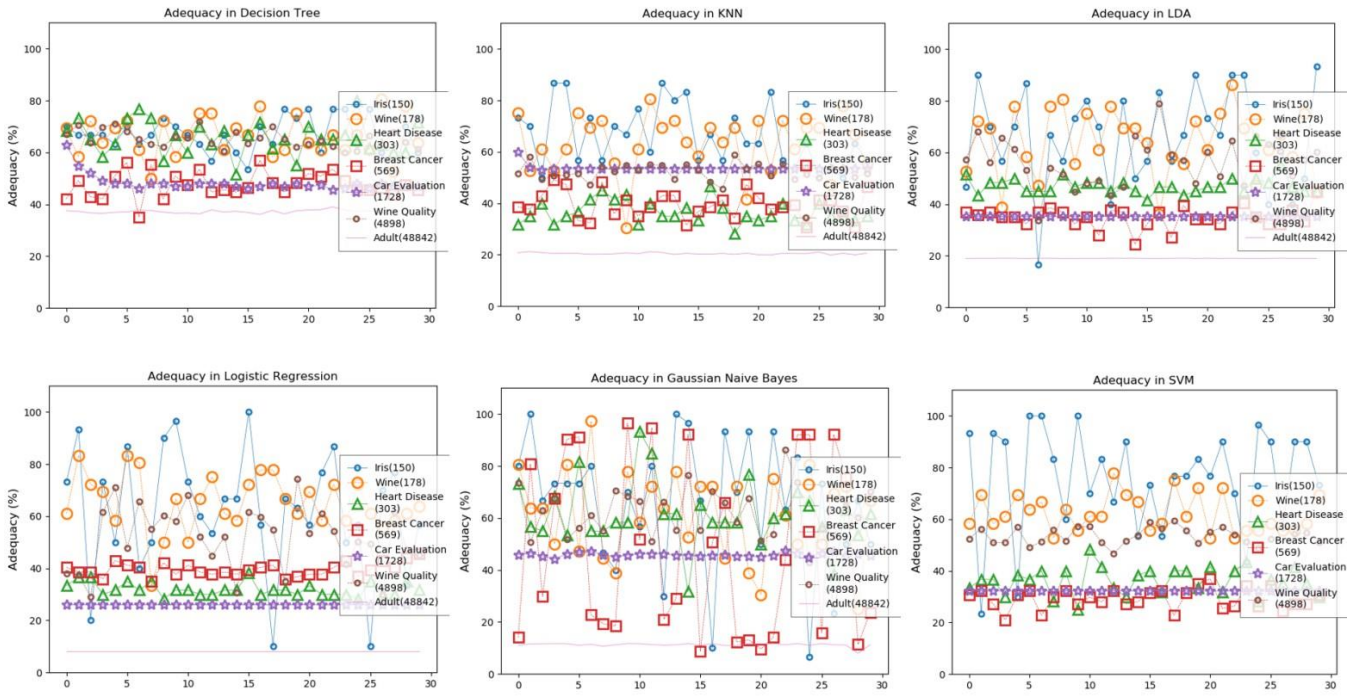

Fig. 2. Adequacy distribution for each mutant

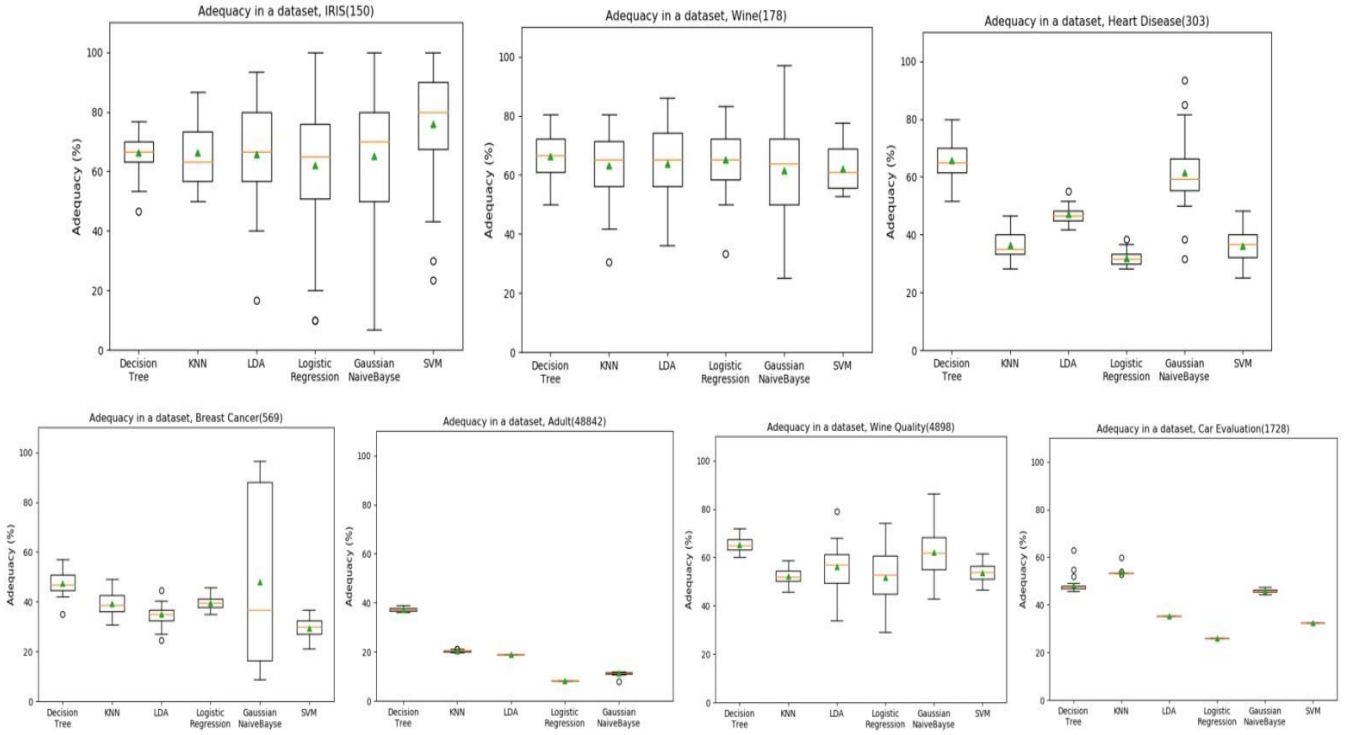

Fig. 3. Distribution of adequacy for each dataset

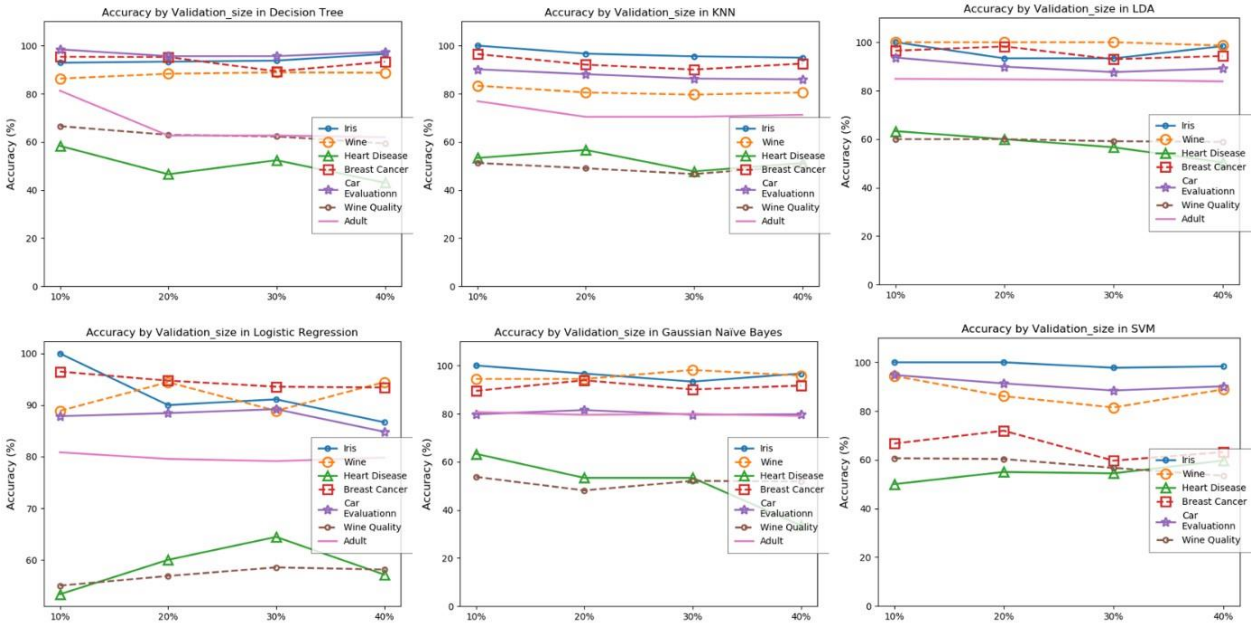

Fig. 4. Accuracy distribution for validation size