MDLE Work 1 - Reproducibility

Diogo M. Marto NºMec: 108298*

University of Aveiro, Portugal

1. Replicating the results

For this work, the objective was to replicate the results of obtained in [1] and point out possible improvements to improve the replicability of the results. All the code used for this work is in [2].

1.1. Dataset Description

The dataset provided for this work ('TUANDROMD.csv') had 4465 rows × 242 columns, which is a different dimension than the one provided in the paper, so the results I obtain **may** differ by lot if I had the one pointed out exactly on the paper. The dataset had 240 features, all of which were binary classes and labels of either 'goodware' or 'malware'. The class distribution of the labels was 80-20 in favour of 'malware', so metrics like accuracy might lose their meaning. Since the paper presented results solely focused on accuracy, I will show the accuracy, but I focused on F1 score and AUC as metrics since they take into account the imbalance.

1.2. Preprocessing

From the paper, it wasn't clear if they made a test set; they only mentioned that they used K-fold cross-validation with 10 folds. To address this ambiguity, I trained the classifiers with 2 different training sets:

- The entire dataset with k-fold cross-validation used both as training and test. The results are the average of the metrics across the folds.
- And a more correct way, where I created test and train set with a 20-80 split and the classifiers were trained k-fold cross-validation only on train set and then the metrics were computed on the test set.

I also assumed that normalization of the features wasn't required since they were all classes and had the same range of values (either 0 or 1), so there wasn't a dominant feature.

1.3. Model creation

As mentioned in the previous section, I have 2 different training sets, so naturally, I also have 2 ways to train the models. Besides, I also decided to include Logistic Regression and Support Vector Machines in the models because I found it odd that in the paper, they only used ensemble classifiers, and based on some literature [3] I found out that Logistic Regression and Support Vector Machines are somewhat common in problems of this area. Since no mention of hyperparameters is found in the paper, I choose 2 approaches:

- Using the entire dataset has training with cross-validation and training the classifiers with default parameters. This was done to simulate how a not-so-experienced person would train these models.

*Corresponding author. E-mail address: diogo.marto@ua.pt

- Using test and train sets and training the classifiers on a grid search with cross-validation 10 and choosing the best model based on the F1-score of the average validation score and then obtaining the final metrics with the test set.

Both approaches confirmed more or less the results obtained in the paper. The metrics I obtained were slightly higher, but this could be due to the different dataset from the one shown in the paper. Both Logistic Regression and Support Vector Machines showed very good results, and the Support Vector Machines was the model with the best scores, so I don't understand why they only chose ensemble models in the paper.

2. Steps to improve the replication

The most obvious step to improve the replication of the results is to provide the code and the environment used to obtain the results presented, two things the paper didn't do. Besides that, a more thorough justification of the training and evaluation process should be provided in the paper, for example, if they made a test and train set and the hyperparameter choice.

3. Appendix

3.1. Use of AI

In my VScode, I have Github Copilot and used some of its auto-complete features to fill in some code faster. Besides that, I also asked ChatGPT some questions related to scikit-learn for example, "How exactly does scikit-learn do cross-validation".

References

[1] P. Borah, D. K. Bhattacharyya, and J. K. Kalita, "Malware Dataset Generation and Evaluation," in *Proceedings of the Conference on Malware Analysis*, Tezpur, Assam, India, 2025.

[2] D. M. Marto, "MDLE 1." [Online]. Available: https://github.com/DiogoMMarto/MDLE_1

[3] B. Sanjaa and E. Chuluun, "Malware detection using linear SVM," in *Ifost*, 2013, pp. 136–138. doi: 10.1109/IFOST.2013.6616872.