IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A MALWARE CLASSIFICATION APPROACH USING DECISION TREE ALGORITHM

Midun G [1], Ramesh E.R [2]

Midun G (M.Sc., Department of Computer Science and Engineering, Dr. MGR Educational and Research Institute, Chennai, India)

E. R. Ramesh (Faculty, Centre of Excellence in Digital Forensics, Dr. MGR Educational and Research Institute, Chennai, India)

Abstract:

There are numerous attack types a hacker may carry out these days. One of the popular ones is malware, which can be malicious code or a section of code. It is usually created for the purpose of damaging an operating system of a device. The python programming is the most common form of malware and it is in the form of virus, worm, trojans, ransomware, spyware, adware and more. The machine learning technique is used by the python programming to classify malware samples using a decision tree classifier. It is a kind of supervised learning algorithm that is responsible for data pre-processing first and then identifying experience features. The model is trained on a training dataset and evaluated using a testing dataset, as it parted the data into two parts – for testing and training. Classifier using a decision tree works by recursively dividing feature space, choosing the node on which to base the decision to only consider feature values at each node. Generally, the classifier runs with the default hyperparameters, but these parameters can be adjusted to suit specific needs. The model being evaluated uses an f1-score metric that's convenient to identify what it can classify it as malware (1) or benign (0). The dataset undergone in training and testing of dataset and it show the 99% of accuracy with the decision tree algorithm.

Keyword:

Malware, Benign, Decision tree, Accuracy.

I. INTRODUCTION

Malware, short for malicious software or malicious code refers to any software specifically designed to disrupt, damage, or gain unauthorized access to computer systems, networks, or devices. Malware can take various forms, including viruses, worms, trojans, ransomware, spyware, adware, and more. Its primary intent is often to steal sensitive information, corrupt data, or cause system malfunctions, all without the consent or knowledge of the user. Malware is typically created by cybercriminals with malicious intent and distributed through various means, such as email attachments, malicious websites, or compromised software.

In today's world, where cyber threats are increasingly becoming sophisticated, malware detection and classification are some of the most important issues in maintaining the security of computers and networks. Traditional methods, which are signature-based, generally cannot detect newly born malware variants. Hence, there arises a need for more advanced techniques in the detection of malware. Machine learning algorithms, including but not limited to Decision Trees, provide an excellent way to identify the type of malware by their behavior patterns and characteristics. In this paper, we have experimented with Decision Trees for the classification of malware and have reported the results.

2 Research Methodology

This Research work uses python as a programming language and it import the required elements, then it take the dataset which we have taken from Kaggle site and it consist of 19612 data about the malware. It then removes the elements which have the no data content and which have a unique value will be not consider on the classification, so it drops certain and maintain the rest in the classification process

There was a total of 19612 content in the dataset in that 70% is been used for training (13728) and the rest (5884) is used for testing the data. The testing data is tested using classification method.

2.1 Data loading and exploration

It start from the importing of required libraries and loading the dataset, the dataset is take from the site of Kaggle and it contain of malicious code that file is known as CSV (comma-separated value) file. After loading the files it identifies the unique values in the dataset and print it on the output are and that things won't be in the part pre-processing as they have same values so it will be dropped

2.2 Data Preprocessing:

Although irrelevant columns like 'Name', 'Malware', etc. are dropped from the DataFrame to prepare the feature set (x) and target variable (y), you can be sure that this is done to ensure the quality of your data. The dataset is divided into training and testing sets using the train_test_split function from scikit-learn, a human-like way to show how this important step helps to build a better model.

2.3 Model Training:

A Decision Tree Classifier is created (dt = DecisionTreeClassifier()) . The classifier is trained using the training data (x_train, y_train) with the fit method.

2.4 Model Evaluation:

The predictions are made on the test set using the trained model which is excellent for accurate prediction. predict(x_test)). The classification report shows what the model thinks it is doing, with the help of the classification_report function from scikit-learn, which gives you numbers like, the ratio of good guesses, the ratio of wrong guesses, and the ratio of both good and wrong guesses.

2.5 EvaluationMetrics

- Precision: It is the ratio of the count of truly predicted positives to the sum of all positive predictions. It gives an indication of how accurate the positive predictions are.
- Recall: The ratio of true positive predictions to the total number of actual positives. It measures the model's ability to identify all relevant instances.
- F1-Score: The harmonic mean of precision and recall. It gives a single metric for evaluating the model's performance by balancing precision and recall.

Flow chart:

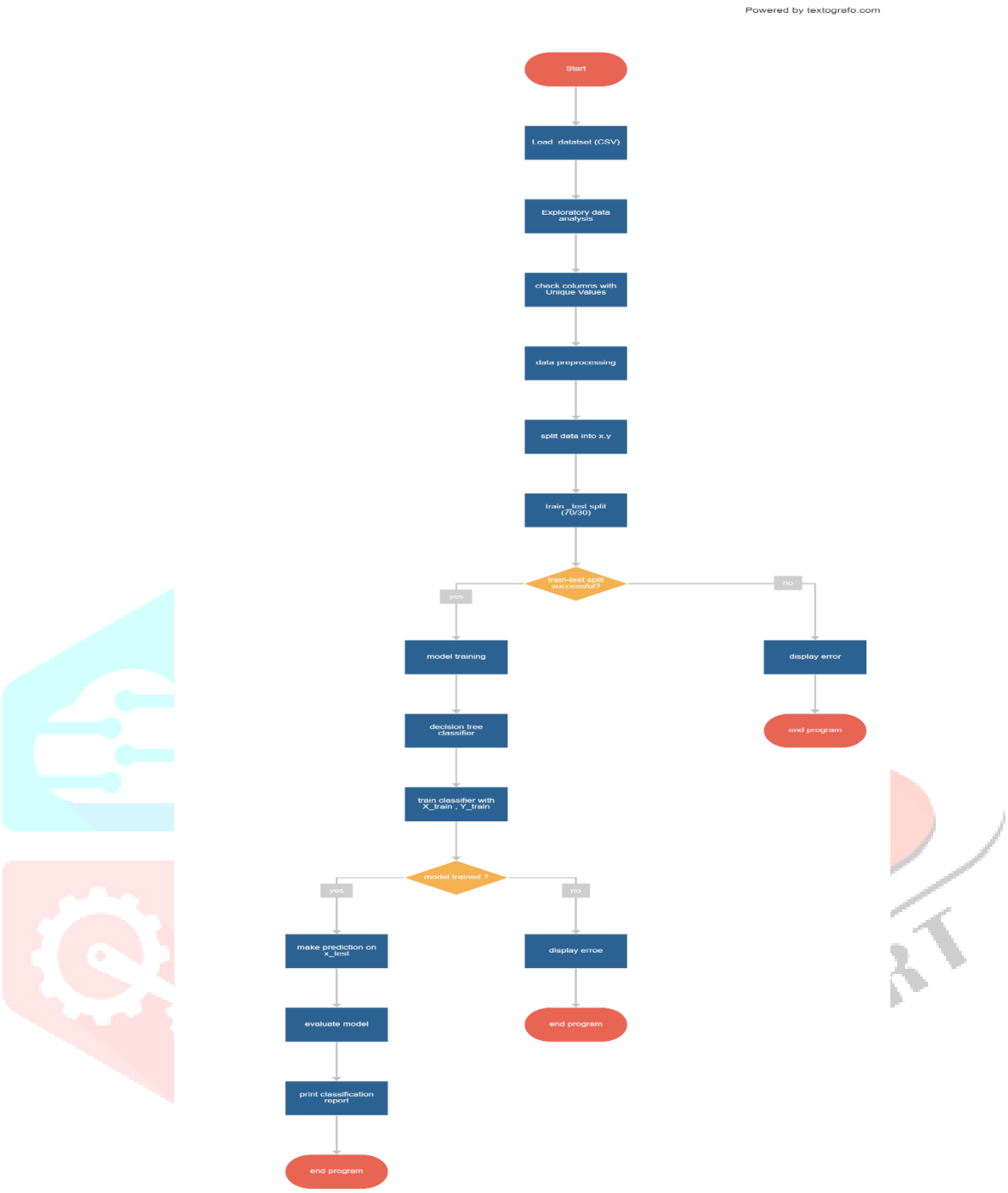

Fig 1 flow drawing of the coding

Results

The classification report generated after the execution

	precision	recall	f1-score	support
0	0.98	0.97	0.98	1538
1	0.99	0.99	0.99	4346
accuracy			0.99	5884
macro avg	0.98	0.98	0.98	5884
weighted avg	0.99	0.99	0.99	5884

Fig 2 output of the classification model

Accuracy: 99% of the samples in the test set were correctly classified.

Precision about Benign: 0.98; in other words, 98% of the samples predicted as being benign actually were benign.

Recall for Benign: 0.99, which indicates that 99% of the actual benign samples are well classified.

Precision in malware: 0.97. It shows the portion of the samples that are really malware, out of all the malware predictions.

Recall for Malware: 0.99; which indicates that 99% of the real malware samples were correctly identified

Conclusion

This paper has provided a technique to categorize malware samples employing a Decision Tree algorithm. The method shows good results in correctly recognizing malware just using their features. we can design models that are capable of detecting and categorizing the malware regardless of their respective features by analyzing the file type and remainder of the files. The decision tree can be easily comprehended, thus, insights into the decision-making is possible. This leads to more predictability and accurate predictions of the intended outcomes. Further research may include fine tuning the classifier and application of ensemble methods to enhance type performance. Generally, the proposed approach seems to be quite helpful for cybersecurity arena and can be employed for malware detection and mitigation in real time scenario

References

1. Scikit-learn documentation. Retrieved from https://scikit-learn.org/stable/documentation.html
2. Pandas documentation. Retrieved from https://pandas.pydata.org/docs/
3. Akshay Zadgaonkar, Ravindra Keskar, Omprakesh Kakde, Towards a Machine Learning Model for Detection of Dementia Using Lifestyle Parameters 24 sept 2023
4. C. Muller and S. Guido, Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, 2016.
5. J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, 2016.
6. J. Han, J. Pei, and M. Kamber, Data Mining: Concepts and Techniques. Elsevier, 2011.
7. J. R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.
8. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Wadsworth International Group, 1984.
9. J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 470-478.
10. M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data Mining Methods for Detection of New Malicious Executables," in Proceedings of the IEEE Symposium on Security and Privacy, 2001, pp. 38-49.
11. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.