AndroHealthCheck: A Malware Detection System for Android Using Machine Learning

Prerna Agrawal and Bhushan Trivedi

Abstract With the boom of malware, the area of malware detection and the use of gadget assist to gain knowledge in research drastically with the aid of researchers. The conventional methods of malware detection are incompetent to detect new and generic malware. In this article, a generic malware detection process is proposed using machine learning named AndroHealthCheck. The malware detection process is divided into four phases, namely android file collection, decompilation, feature mining and machine learning. The overall contributions made in AndroHealthCheck are as follows: (1) designing and implementing a crawler for automating the process of benign files download, (2) collection of unstructured data from the downloaded APK files through the decompilation process, (3) defining a proper mechanism for the feature selection process by performing a static analysis process, (4) designing and implementing a feature mining script for extracting the features from unstructured data collection from APK files, (5) generating a rich homemade data set for machine learning with a huge variety and different flavours of malware files from different families and (6) evaluating the performance of the generated data set by using different types of supervised machine learning classifiers. In this article, the overall architecture and deployment flow of AndroHealthCheck are also discussed.

Keywords Malware detection · APK files · Static analysis · Unstructured data · Feature mining · Machine learning

1 Introduction

The malware detection domain using machine learning is an emerging area that is being researched extensively these days. The conventional methods used for the

P. Agrawal (✉) · B. Trivedi
Faculty of Computer Technology, GLS University, Ahmedabad, Gujarat, India
e-mail: prerna.agrawal@glsuniversity.ac.in

B. Trivedi
e-mail: bhushan.trivedi@glsuniversity.ac.in

A. Pasumpon Pandian et al. (eds.), *Computer Networks, Big Data and IoT*, Lecture Notes on Data Engineering and Communications Technologies 66,
https://doi.org/10.1007/978-981-16-0965-7_4

detection of malware are more resource and time consuming and are incompetent to detect generic and new malware [1]. The conventional methods used for malware detection include signature-based, resource-based, components-based and permission-based analysis [2], which are not enough to detect the new and generic malware. Machine learning methods acquire on their own from the knowledge given to them as training data and use performed classification on testing data and are highly used for the investigation of the malware [1].

Here, the Android files are used as a proof of concept for the proposed malware detection process. For the proper investigation of the malware, the independent flavours of features and a variety of malware files from different malware families are needed. The existing malware data sets are available and that can be used in machine learning directly, but the Drebin data set is found to be with lesser features and with malware files having less variety of malware families. To generate our data set with independent flavours of features, they were decided to have a huge variety of malware files for better performance in malware detection. It can be directly used by researchers in machine learning.

The overall contributions performed in AndroHealthCheck are as follows: (1) designing and implementing a crawler for automating the process of benign files download [3], (2) collection of unstructured data from the downloaded APK files through the decompilation process [4], (3) defining a proper mechanism for the feature selection process by performing a static analysis process [5], (4) designing and implementing a feature mining script for extracting the features from unstructured data collection from APK files [5], (5) generating a rich data set for machine learning with a huge variety and different flavours of malware files from different families [5] and (6) evaluating the performance of the generated data set by using different types of supervised machine learning classifiers [6]. Using the machine learning classifiers, the performance of the CatBoost classifier is highest with 93.15% accuracy and ROC value of 0.91 [6]. The layout of this article is divided into the following sections. Section 2 describes the overall architectural flow of the AndroHealthChecka—malware detection system. Section 3 describes the overall deployment flow of the AndroHealthCheck—a malware detection system. Section 4 describes the conclusion of the research work.

2 Architecture of AndroHealthCheck

This section represents the overall architecture of AndroHealthCheck—the malware detection system. Figure 1 represents the overall architecture of AndroHealthCheck. The AndroHealthCheck is divided into four phases: They are android file collection, decompilation, feature mining and machine learning. In the android file collection [3] phase, the malware and the benign file collection were concentrated as it is the first step for the data collection. In the android file collection phase, the user enters the website URL for downloading the files. This module connects to the website, and after the successful establishment of connection, the website sends the file request

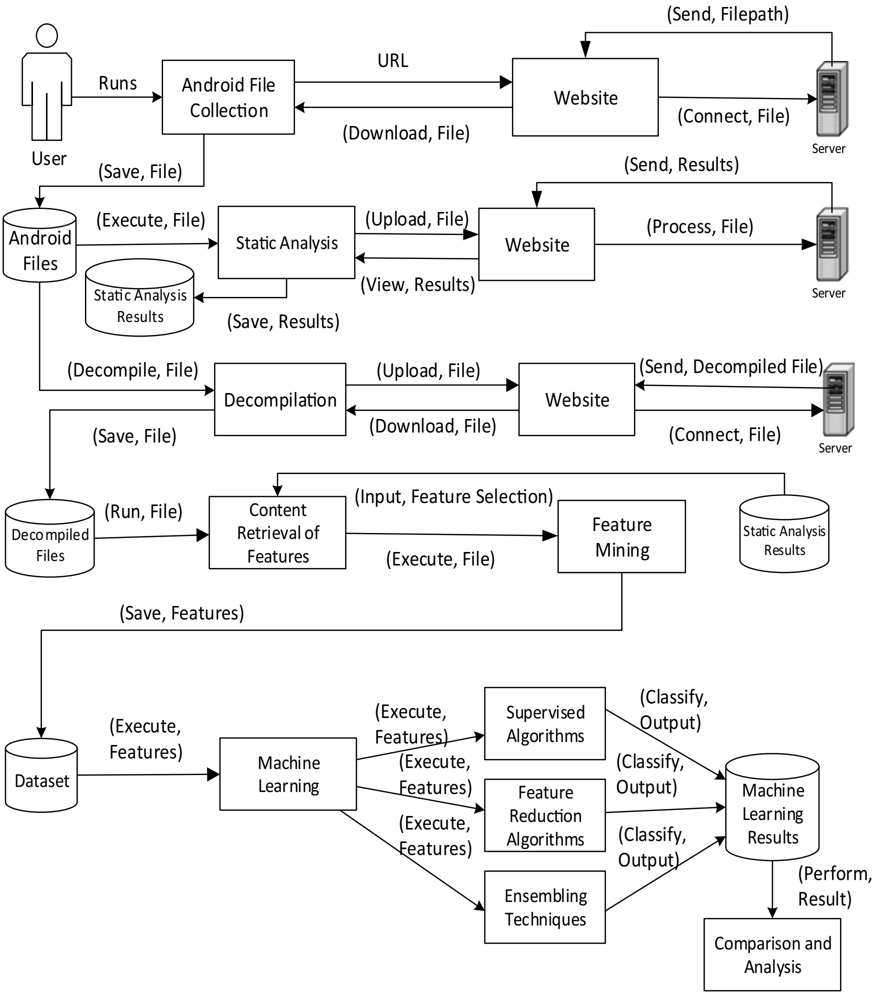

Fig. 1 Overall architectural flow of AndroHealthCheck

to the server for download. The server responds with the file path to the website, and the file gets downloaded and stored in the physical location. Using the android file collection phase [3], a total of 15,506 files of malware files were downloaded from the world's famous android malware projects like Drebin, Androzoo, AndroPRAGuard, Kharon and Kudoos [3]. To automate the process of benign file downloads, a crawler is developed [3] and downloaded for 4000 benign files [3]. The android files contain an unstructured data format like text files, Java files and.xml files. For extracting the features from the APK files, reverse engineering of these files is necessary. So the decompilation phase [4] collects the unstructured data from the APK files. In the decompilation phase, an APK file is given as input. The APK file is uploaded on the website and sent to the server for decompilation. The server processes the APK file, decompiles it and sends the decompiled file back to the website in the form of a zip file. The decompiled zip file is saved to a physical location. Using the decompilation phase, the collected malware and benign files are decompiled, and unstructured data like XML and Java files are collected from each decompiled APK file [4].

For mining, the features from the decompiled files feature selection are an important criterion as there is no proper mechanism available for the feature selection process. So for the proper selection of features, the static analysis [5] was performed using the online malware scanners [7]. In the static analysis phase [5], an APK file is given as an input to the website, and the file is uploaded on the website. The file is sent to the server for processing. The server processes the file and returns the results to the website. The user can view the results of the file, and those results are saved to a physical location. All the collected APK malware files were scanned using the online malware scanners, and their reports were collected and analysed. From the analysis of the reports of the online malware scanners, total of 215 features were selected that included various permissions, API calls and Intents. In the feature mining, all the 215 selected features were extracted from the unstructured data collection from the APK files. For the feature mining process, a feature mining script was developed and implemented to extract features from the decompiled APK files. In the feature mining phase [5], the feature mining script looks for a decompiled file in the decompiled files repository and extracts all the 215 features from a decompiled file. A vector for each Android file is generated with extracted features and will be saved in a CSV file. Using this feature mining phase [5], a final data set is generated with a total of 16,300 records in which it includes both malware and benign files [5]. For the evaluation and performance of the generated data set, various supervised machined learning classifiers were implemented in the machine learning phase [6]. In the machine learning phase [6], the features from the generated data set were given as input, and various supervised machine learning classifiers were applied to the features for the classification of the malware and benign files. Different types of supervised classifiers, feature reduction classifiers and ensembling techniques were applied, and the classification result of each classifier is saved into an Excel file. The classification results of each classifier are compared and analysed for better performance and detection of malware. The machine learning phase is explained in our previous paper [6]. The next section describes the overall deployment flow of the AndroHealthCheck system.

3 Deployment of AndroHealthCheck

This section discusses the overall deployment flow of the AndroHealthCheck malware detection system. Figure 2 represents the overall deployment flow of AndroHealthCheck. The AndroHealthCheck is divided into four phases such as android file collection, decompilation, feature mining and machine learning. The deployment scenario of the AndroHealthCheck model is discussed according to its phases.

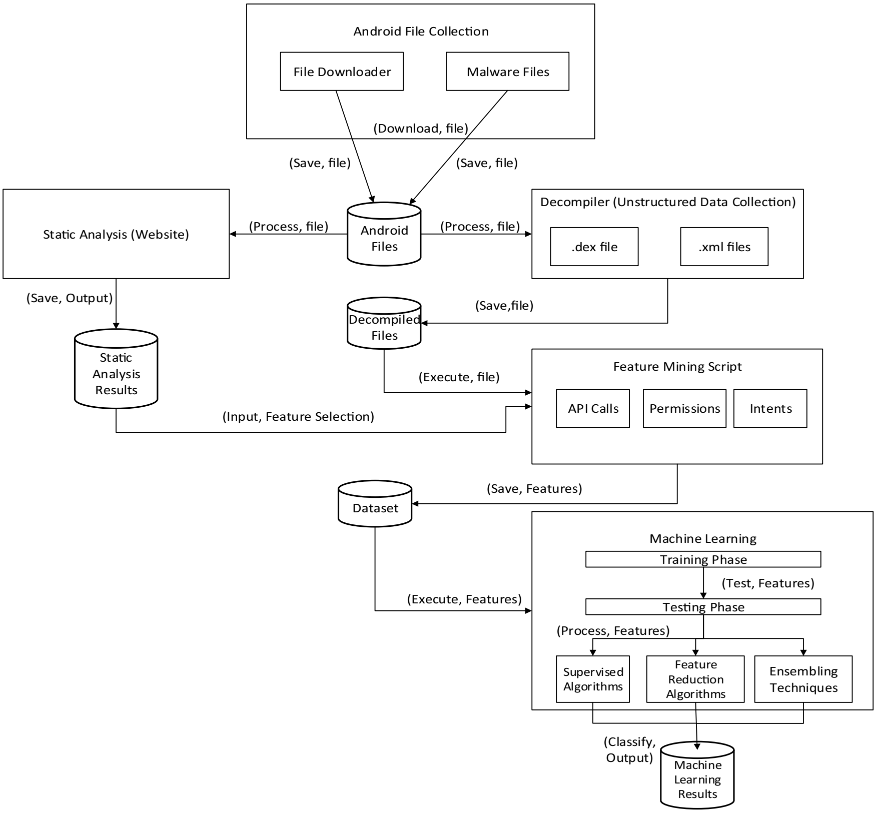

Fig. 2 Overall deployment flow of AndroHealthCheck

In the android file collection phase, the malware files were downloaded from different android malware projects. A variety of malware files are downloaded from different cloud servers and online sources. For benign files download, we have designed and implemented a crawler/file downloader to automate the process of automatic file download from websites [3]. The crawler is designed using technologies like node package and Cypress framework. The version of Node 10.16.3 is used with Cypress 4.0 along with Chrome browser. The crawler can be deployed on any local machine and executed using Chrome browser. The crawler needs to be initialized with a website URL, and it will fetch and parse a web page by downloading the APK files. The malware files and benign files downloaded are saved into android files repository. In the static analysis phase [5], different online malware scanners were used to process the file and to obtain the results. The downloaded APK file is given as an input to the website, the file processing is done on the cloud servers itself, and the processed zip file is returned to the website and downloaded at a physical location. In the decompilation phase [4], an online decompiler is used for the collection of unstructured data from the APK files. The APK file is given as an input to the website, and the file is uploaded on the server and is decompiled on the cloud server only. The decompiled file is returned to the website back and saved to a physical location. The decompiled file contains .dex files and .xml files. The .dex files are further processed to extract the Java files.

In the feature mining phase [5], the selected 215 features from analysing the reports of the static analysis phase are extracted from the decompiled APK files. A feature mining script is developed and implemented in Python for extracting the features from the decompiled files. The feature mining script can be deployed on any local machine and executed using Python, and features can be extracted from the decompiled files and saved in a CSV file. The feature mining script mainly extracts API calls, permissions and Intents from the APK files. A vector of extracted features is generated for each Android file and saved in the CSV file. Using the feature mining script a final data set of a total of 16,300 records is generated. For the performance and evaluation of the generated data set of various machines, learning classifiers are used. In the machine learning phase [6], various experiments are carried on the data set by using various machine learning classifiers. The experiments are carried on Intel Core i7-7500U CPU @ 2.90 GHz with 8 GB RAM and Windows 10. The technologies used for experiments are Python and Anaconda Package having a suite of various machine learning libraries for supervised and unsupervised algorithms. For obtaining good results, the data set is divided into a training set and a testing set. From the data set, 75% of data is trained to perform classification and testing on 25% of data. Various machine learning classifiers applied are KNN, random forest, decision tree, linear SVM, logistic regression, Naive Bayes, linear discriminant analysis (LDA), non-negative matrix factorization (NMF), principal component analysis (PCA), bagged decision tree, extra trees and random forest using bagging, gradient boost, CatBoost, AdaBoost, XGBoost, softmax voting and hardmax voting. The classification results of all the machine learning classifiers are stored in an Excel file.

4 Conclusion

AndroHealthCheck is a malware detection system to investigate that a file is malware or not by using machine learning methods. A data set is a prerequisite for machine learning models to determine themselves and gain knowledge from the training data for the proper classification of malware and benign files on the testing data. For the investigation of proper malware, a homemade data set with rich variety and with different flavours of malware families was needed. The objective is to create our own data set which can be directly used by researchers for machine learning. A generic malware detection process named AndroHealthCheck is proposed for malware detection using machine learning.

The AndroHealthCheck defines the generic process of data set generation for machine learning and also defines a mechanism for malware detection using machine learning. The architectural and deployment flow of AndroHealthCheck were discussed. In AndroHealthCheck, the design and implementation of a crawler for automating the process of benign files download were discussed. The unstructured data were collected from the APK files through the decompilation process from all downloaded 15,506 malware and 4000 benign APK files. A proper mechanism is defined for the feature selection process from the APK files through static analysis. The design and implementation of a feature mining script are used for feature mining from unstructured data collection from APK files. A rich data set is generated for machine learning of a total of 16,300 records and 215 features with a huge variety and different flavours of malware files from different families and independent flavours of features. The performance of the generated data set is evaluated with different supervised machine learning classifiers and found that the performance of the CatBoost classifier is highest with 93.15% accuracy and ROC value of 0.91.

Acknowledgements We would like to acknowledge our students Ms. Indushree Shetty, Ms. Sabera Kadiwala, Mr. Yash Gajjar, Mr. Ronak Jain, Mr. Vraj Shah and Mr. Akshay Ardeshana of GLSICT, GLS University, who helped us immensely in different phases of AndroHealthCheck a malware detection system for Android.

References

1. Agrawal P, Trivedi B (2020) Machine learning classifiers for android malware detection. In: 4th International conference on data management, analytics and innovation (ICDMAI). Springer AISC Series, New Delhi, pp 311–322. https://doi.org/10.1007/978-981-15-5616-6_22.ISBN 978-981-15-5616-6
2. Agrawal P, Trivedi B (2019) A survey on android malware and their detection techniques. In: Third international conference on electrical, computer and communication technologies (ICECCT) IEEE, Coimbatore. https://doi.org/10.1109/ICECCT.2019.8868951,E-ISBN 978–1–5386–8158–9
3. Agrawal P, Trivedi B (2020) Automating the process of browsing and downloading APK files as a prerequisite for the malware detection process. Int J Emerg Trends Technol Comput Sci (IJETTCS) 9(2):013–017. ISSN 2278-685
4. Agrawal P, Trivedi B (2020) Unstructured data collection from APK files for malware detection. Int J Comput Appl (IJCA) 176(28):42–45. https://doi.org/10.5120/ijca2020920308.ISBN 973-93-80901-12-5, ISSN 0975 – 8887,
5. Agrawal P, Trivedi B (2020) Feature mining from APK files for malware detection. Int J Appl Inf Syst (IJAIS) 12(32):6–10. https://doi.org/10.5120/ijais2020451874. ISBN 973-93-80975-75-9, ISSN 2249 - 0868
6. Agrawal P, Trivedi B (2020) Evaluating machine learning classifiers to detect android malware. In: IEEE International conference for innovation in technology (INOCON), Bangalore (Paper Selected)
7. Agrawal P, Trivedi B (2019) Analysis of android malware scanning tools. Int J Comput Sci Eng (IJCSE) 7(3):807–810. https://doi.org/10.26438/ijcse/v7i3.807810,E-ISSN 2374–2693