Android Malware Classification Using K-Means Clustering Algorithm

Isredza Rahmi A Hamid, Nur Syafiqah Khalid, Nurul Azma Abdullah, Nurul Hidayah Ab Rahman, Chuah Chai Wen

Information Security Interest Group (ISIG), Faculty Computer Science & Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.

Corresponding author: rahmi@uthm.edu.my, nursyafiqahkhalid@gmail.com, azma@uthm.edu.my, hidayahar@uthm.edu.my, cwchuah@uthm.edu.

Abstract. Malware was designed to gain access or damage a computer system without user notice. Besides, attacker exploits malware to commit crime or fraud. This paper proposed Android malware classification approach based on K-Means clustering algorithm. We evaluate the proposed model in terms of accuracy using machine learning algorithms. Two datasets were selected to demonstrate the practicing of K-Means clustering algorithms that are Virus Total and Malgenome dataset. We classify the Android malware into three clusters which are ransomware, scareware and goodware. Nine features were considered for each types of dataset such as Lock Detected, Text Detected, Text Score, Encryption Detected, Threat, Porn, Law, Copyright and Moneypak. We used IBM SPSS Statistic software for data classification and WEKA tools to evaluate the built cluster. The proposed K-Means clustering algorithm shows promising result with high accuracy when tested using Random Forest algorithm.

1. Introduction

Malware is developed to gain an access or damage computer without the user's knowledge. There are many cases of malware such as spyware, key loggers, or viruses that affect organization data processor [1]. Malware continues to grow and evolve to bypass antivirus and other levels of protection, which makes it hard for security team to keep up. More than 4,000 ransomware attacks have occurred every day since year 2016 [18]. That is a 300% increase over year 2015, where 1,000 ransomware attacks were seen per day. Through malware, criminals are able to infect large numbers of victims at once by automating these attacks and extend the reach of their infections to multiple systems per victim. This can cause more damage and potential downtime which put more pressure on victims to resolve the issue quickly. Commonly, people stored important data on electronic devices such as laptop and mobile device without making any backup. Once the electronic devices being infected or attacked by Android malware, it is difficult to retrieve the data back.

There are two types of Android malware which are Ransomware and Scareware. Ransomware exploded into a billion-dollar industry in 2016 that create a gold-rush atmosphere for cyber criminals, with demand for and supply of new ransomware variants and delivery platforms [19]. Ransomware works through spam email which contains malicious attachment. The malicious attachment asked the user to open the attachment with a convincing appearance. Once infected, ransomware prohibits or

limits the user from accessing the system either lock the computer's screen or encrypt file that had been typeset with a password [6]. Then, ransom message is displayed which instruct the user to pay ransom money through payment system such as Ukash or Paysafecard [2] in order to have the access again. Conversely, scareware is known as fake anti-virus software which becomes the most common methods to deceive the victim's money. Microsoft detected scareware approximately 52 million times in United States in year 2011 [7]. The scareware program looks similar with the legitimate security programs. Normally, the scareware claimed that it has detected a large number of nonexistent threats on the computer and then urge the victim to pay for full version of the software to remove the threats.

This paper focus on Android malware classification using K-Means clustering algorithm tested on two datasets extracted from ransom.mobi detector [3]. Virus Total dataset consists of 907 samples while Malgenome dataset consists of 1255 samples. Both datasets have nine types of features which include Lock Detected, Text Detected, Text Score, Encryption Detected, Threat, Porn, Law, Copyright and Moneypak. Then, the Android malware class which is build using K-Means clustering algorithm will be analysed using Random Forest algorithm [4]. The objectives of this paper are:

a) to design an Android malware classification model based on behaviour approach.
b) to classify the Android malware using K-Means clustering algorithm.
c) to evaluate the proposed model in terms of accuracy using machine learning algorithms.

The rest of the paper is organized as follows: Section 2 describes the related work on Android malware classification and K-Means clustering technique. Section 3 presents the proposed classification model for Android malware classification where each cluster prediction becomes elements of the cluster. The cluster constructed from rule-based clustering algorithm is then used to train the classifier algorithm. Section 4 shows the performance analysis evaluation methodologies and experimental results. Finally, Section 5 concludes the work and highlights a future research.

2. Related Work

Malware can be in various forms of code, scripts, active content and other software. It is a universal term applied referring to several kind of hostile software includes computer viruses, ransomware, worms, Trojan horses, rootkits, key loggers, dialers, spyware, adware and other harmful programs [5].

2.1. Android Malware Classification Approach

Several anti-ransomware techniques have been proposed in recent years to detect and prevent the increasing number of ransomware attacks as shown in Table 1. In general, many researchers [11][12][13] apply the clustering algorithm in order to classify Android malware. Table 1, shows the comparative analysis on Android malware classification approach. Work by Wu et al. [11] proposed DroidMat to detect Android malware using behaviour-based features. DroidMat extracts the static information from each application's manifest file and API Calls related to permissions. K-means algorithm is applied to enhance the malware modelling capability. Then, the number of clusters is determined by Singular Value Decomposition (SVD) method on the low rank approximation. Finally, it exploit k-Nearest Neighbour (kNN) algorithm to classify the application as benign or malicious. They manage to achieve 97.87% accuracy tested on Contagio Mobile dataset.

Work by Burguera et al. [12] proposed a dynamic analysis of application behavior for detecting malware in the Android platform (Crowdroid). The Crowdroid is set in the framework to collect traces from real users based on crowd sourcing. They achieved 100% accuracy tested on two types of data sets: artificial malware created for test purposes, and real malware from Virus Total. However, the experiment was tested for small amount of data. Other work by Aung et al. [13] implemented a framework for classifying Android applications using machine-learning techniques. This system monitors various permission based features and events obtained from the Android applications. They tested on 200 samples of dataset using machine learning classifiers to classify whether the application is benign or malware. Our work differ than Aung et al. [13] in such a way that, we classify Virus Total and Malgenome dataset using K-means algorithm into three categories; ransomware, scareware or goodware. Moreover, work by Schlesinger et al. [8] used live data with permission-based feature

where we used behavior-based feature. Then, we grouped the virus using K-Means clustering algorithm. We choose Random Forest algorithm because it is most suitable algorithm on both datasets.

Table 1: Comparative analysis on Android Malware Classification Approach

Work By	Features	Algorithm	Dataset	Result
DroidMat [11]	permissions, deployment of components, Intent messages passing and API calls	K-Means and KNN	Contagio Mobile	97.87%
Crowdroid [12]	Behavior-based Android malware	K-Means	Virus Total	100%
Permission-based [13]	Permission and event	K-means and Random Forest	Android Application	91.75%

2.2. Classification Data

Generally, there are two types of classification data; Unsupervised and Supervised Learning. Unsupervised learning did not provide the model with the correct results during the training. Therefore, the basis of their statistical properties can be used to be the cluster only. The cluster can be carried out even if the class are only available for a small number of objects representatives of the desired classes [8]. On the other hand, supervised learning provides the training input data with the desired results. The correct results are known and given as an input to the model during the learning process. The construction of proper training, validation and test set are very crucial. These methods are usually fast and accurate. Besides, it have to be able to generalize which give the correct results when new data are given in input without knowing a priori the target [8].

3. Classification Model

This section explains about the classification model using K-Means clustering algorithm.

3.1. Android Malware Classification Model

There are five phases needed to classify the unsupervised data which are raw data, pre-processing, feature extraction, clustering algorithm and classification algorithm as shown in Figure 1. We classify the Android malware into three types that are ransomware, scareware and goodware. Two dataset were extracted from ransom.mobi detector [3]; Virus Total and Malgenome. These datasets are unsupervised data which is used for exploratory data analysis to find hidden patterns or group of data. The preprocessing data is data mining techniques that transform the raw data into an understandable format. To complete this phase, raw data must go through a series of preprocessing steps in Table 2.

Table 2. Steps in pre-processing

Preprocessing Steps	Description
Data Cleaning	Fill in missing values, smooth the noisy data or resolve inconsistencies in the data.
Data Integration	The conflict within the data will resolved as the data with different representation are put together.
Data Transformation	Data is normalized, aggregated and generalized.
Data Reduction	Present a reduced representation of the data in a data warehouse.
Data Discretization	Involves the reduction of a number of values ranges of attribute intervals.

3.2. Feature Extraction

Initially, the Virus Total and Malgenome dataset were downloaded from ransom.mobi detector website [3]. We selected 907 and 1255 sample of Virus Total and Malgenome dataset respectively. The file is decompressed to extract the necessary Android malware features in .xls format. Both selected dataset have nine features which are used to classify Android malware according to its class. The features are; Lock Detected, Text Detected, Text Score, Encryption Detected, Thread, Porn, Law, Copyright and Moneypak [3]. However, behavior of ransomware virus can be profile based on three features such as Locking Detector, Encryption Detector and Threatening Text Detector [4] as shown in

Table 3. Then, we built dataset in (.arff) file format from the extracted features. Finally, we tested the dataset using Random Forest classification algorithm to distinguish either the Android malware is Ransomware, Scareware or Goodware because it is more robust. Random Forest algorithm is the combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

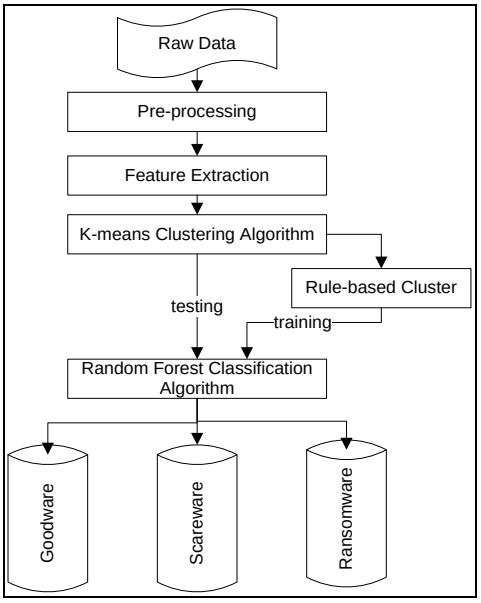

Figure 1: Android Malware Classification Model

Table 3: Behaviour-based Android malware features

Technique	Descriptions
Locking Detector	1. Asking for device-administration right and then lock then device. 2. Superimpose a full-screen alert dialog or activity. 3. Trapping key pressure event such that the "lock screen" cannot be switch away by the victims.
Encryption Detector	1. Encryption key was hardcoded 2. Generates per-device key.
Threatening Text Detector	Notice families localized in English rely on Moneypak for payments, whereas families localized in Russian accept credit card.

3.3. K-Means Clustering Algorithm

In this paper, we classify the dataset using K-Means clustering algorithm. K-Means clustering algorithm is a data mining technique that can be applied in order to sort the dataset into three groups that are ransomware, scareware and goodware [14].

$$J(v) = \sum_{i=1}^{C} \sum_{j=1}^{C_i} \left(\|X_i - V_j\| \right)^2$$

Where, $\|Xi - Vj\|$ is Euclidean distance between Xi and Vj. While C_i is the number of data point in the ith cluster and C is the number of cluster center. To assess the accuracy of this method, the clustered data were compared to the labeled data to determine if cases were clustered appropriately. Given a priori knowledge that the current set contained three types of Android malware, the k-means algorithm is set to three clusters. The true or optimal value of k is not easily determined [14].

Let $X = \{X_1, X_2, X_3, \ldots\ldots\ldots\ldots\ldots X_n\}$ be the set of data points and $X = \{V_1, V_2, \ldots V_c\}$ be the set of centers. Then, randomly select 'c' cluster centers. Next, calculate the distance between each data point and cluster centers. After that, specify the data to the cluster center whose distance from the cluster is

the minimum of all cluster centers. Subsequently, recalculate the new cluster center using formula $V_i = (\frac{1}{C_i}) \sum_{j=1}^{C_i} X_1$ where, C_i represents the number of data points in the ith cluster. Finally, recalculate the distance between each point and obtained new cluster centers. Stop if no point was reassigned; otherwise repeat again the process of specifying the data point to the cluster center whose distance from the cluster is the minimum of all cluster centers [20].

3.4. Rule-based Clustering

Table 4 shows rules-based clustering that have been used in order to classify Virus Total and Malgenome dataset belong to the selected clusters. There are two features that must be considered such as Lock Detected and Encryption Detected. If both features are true, the data is ransomware. Moreover, if Lock Detected feature value is true and Encryption Detected is false then the data is scareware. However, if both features show false value, the data is considered as goodware.

Table 4: Rules-based Clustering

Features	**Android Malware**		
	Ransomware	**Scareware**	**Goodware**
Lock Detected	True	True	False
Encryption Detected	True	False	False

4. Performance Analysis

This section presents the experimental setup and performance metric used to classify Android malware.

4.1. Experimental Setup

Initially the experiment was started by collecting the dataset from ransom.mobi detector [3]. We used two types of datasets from Virus Total and Malgenome. Then, the samples were extracted from each dataset and save it as .csv file. After that, the K-Means clustering process was done using IBM SPSS Statistic to cluster the Android malware into three types which are ransomware, scareware and goodware. Next, the predicted clusters that were obtained based on the k-means clustering algorithm were saved in .csv file. Since the dataset is unsupervised data, we run the predicted cluster on the rule-based clustering to classify the Android malware either ransomware, scareware or goodware. Next, we split the supervised data with predicted cluster using 60:40 ratio size where 60% of the dataset will be used as a training set, while 40% will be used as a testing set. Finally, we tested the proposed clustering approach by using Random Forest classification algorithm on Waikato Environment for Knowledge Analysis (WEKA) tools. After processing the model using the training system, the model will make predictions against the test set. The testing set contains values which are known to the attribute that need to predict. Therefore, it is easy to determine whether the conjecture of the model is correct. Once the model has been trained and tested, it needs to measure the performance of the model.

Algorithm: Rule-Based Clustering

```
INPUT: Class
BEGIN
1: FOR (each incoming DATA) DO
2: IF (LockDetected == TRUE && EncryptionDetected == TRUE) THEN
3: GIVE value Ransomware
4: ELSEIF (LockDetected == TRUE && EncryptionDetected == FALSE) THEN
5: GIVE value Scareware
6: ELSEIF (LockDetected == FALSE && EncryptionDetected == FALSE) THEN
7: GIVE value Goodware
8: ENDIF
9: ENDFOR
10: END Rule-Based Clustering
```

Figure 2. Rule-based Clustering Algorithm

4.2. Rule-based Clustering Algorithm

Figure 2 shows the rule-based clustering algorithm. The input to the Rule-based clustering algorithm is LockDetected and EncryptionDetected feature value. In step 2 to 8, each incoming data will mine LockDetected and EncryptionDetected value for all dataset to determine whether the data class is ransomware, scareware or goodware. If both LockDetected and EncryptionDetected value are true, set the data class as ransomware. If LockDetected value true and EncryptionDetected value is false, we set the data class as scareware. Finally, if both LockDetected and EncryptionDetected value are False, we set the data class as Goodware.

4.3. Performance Metric

In order to measure the effectiveness of the proposed clustering algorithm, we used the following four performance metrics. These metrics are:

a) Accuracy (Acc): How many Android malware classes are correctly predicted by the rule-based clustering algorithm?
b) Error rate (Err rate): How many Android malware classes are wrongly predicted by the rule-based clustering algorithm?
c) False Negatives (FN): How many Android malware classes go undetected by the rule-based clustering algorithm?
d) False Positives (FP): How many Android malware classes are misclassified?

The accuracy metrics is very significant to compute the number of correctly classified Android malware using the proposed algorithm. If the accuracy value is high, the performance of the proposed algorithm is very effective in order to class the Android malware. Additionally, both FN and FP metrics are very important in measuring the effectiveness of security justification approaches. For instance, FP could have considerable negative values on the utility of detection and protection algorithm. This is because examining them takes time and resources. If the rate of FP is high, user might disregard them. The error rate metric is important to inspect the over fitting issues. Receiver Operating Characteristic (ROC) on the other hand, is the measure of certainty of the algorithm with the classification made.

4.4. Result and Discussion

This section presents the classification outcome of the K-means clustering algorithms on the extracted features. We tested the dataset using Random Forest classification algorithm with 60:40 ratio size values in terms of accuracy value, mean absolute error, Receiver Operating Characteristic (ROC) and True Positive (TP) and False Positive (FP) rate.

4.4.1. *Accuracy Value*. Figure 3 shows the accuracy value for Virus Total and Malgenome dataset tested using Random Forest classification algorithm. The Virus Total dataset achieved the highest accuracy which is 98.12%. On the other hand, Malgenome dataset only reached 74.70% accuracy value. This shows that Virus Total dataset has more accurate and precise group of Android malware classified by k-means clustering algorithm as compared to Malgenome dataset.

4.4.2. *Mean Absolute Error*. Figure 4 shows the Mean Absolute Error for both dataset. Mean Absolute Error is used to measure how close the class prediction with the outcome. The Malgenome dataset has higher error rate with 25.30% value as compared to Virus Total dataset with 1.88% value. Virus Total has the lowest error rate because the selected features in the dataset meet the positive criteria. A good classification of feature will affect the result produce. Therefore, a good Android malware class contributes the dataset to have low error rate, True Positive (TP) and False Positive (FP) rate.

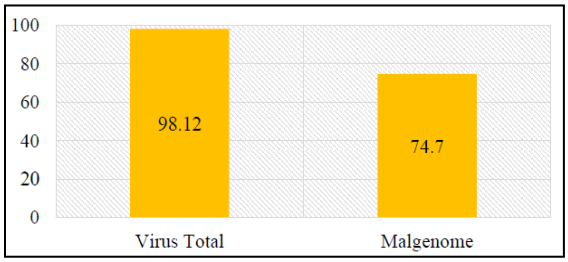

Figure 3. Accuracy value for Virus Total and Malgenome dataset.

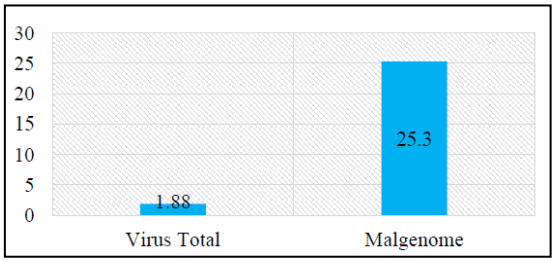

Figure 4. Mean absolute error for Virus Total and Malgenome dataset.

4.4.3. *Receiver Operating Characteristic (ROC).* Figure 5 shows the Receiver Operating Characteristic (ROC) value for Virus Total and Malgenome dataset. The best ROC result is when the value of ROC is near to one. Malgenome dataset has the highest ROC value with 0.997 as compared to Virus Total dataset with 0.994. Both datasets show small discrepancy with just 0.003.

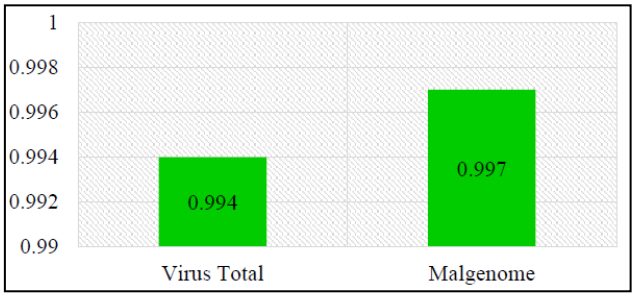

Figure 5. Receiver operating characteristic (ROC)

4.4.4. *True Positive (TP) and False Positive (FP) Rate.* Table 5 shows the True Positive (TP) and False Positive (FP) rate of Virus Total and Malgenome dataset. In order to get the best result, the dataset analysis must obtain the highest TP rate and lowest FP rate. TP value shows that the dataset has correctly classified into its Android malware class. The TP rate for Virus Total and Malgenome dataset are 0.981 and 0.747 respectively. For the FP rate, Malgenome shows the highest values with 0.739 while Virus Total with 0.004.

Table 5. TP and FP rate for Virus Total and Malgenome Dataset

Virus Total		**Malgenome**	
TP	**FP**	**TP**	**FP**
0.981	0.004	0.747	0.739

5. Conclusion

Android malware is an emerging problem nowadays and solving this problem has proven to be very challenging. In this paper, we proposed Android malware classification approach based on K-means clustering algorithm using Lock Detected, Text Detected, Text Score, Encryption Detected, Threat, Porn, Law, Copyright and Moneypak Android malware features as feature vectors. The proposed algorithm is then tested with two dataset; Virus Total and Malgenome dataset. Then, we used rule-based clustering algorithm to group the Android malware into Ransomware, Scareware or Goodware. The proposed rule-based clustering algorithm demonstrates better result when tested on Virus Total dataset with highest accuracy and lowest mean absolute error by 98.12% and 1.88% respectively. However, Malgenome dataset have slightly high ROC value with 0.003 as compared to Virus Total dataset. Overall, Virus Total dataset performed well when tested using the proposed approach with highest TP and lowest FP value. We plan to investigate other Android malware features and datasets practiced on rule-based clustering algorithm to improve clustering accuracy.

Acknowledgement
The authors express appreciation to the University Tun Hussein Onn Malaysia (UTHM). This research is supported by Short Term Grant vot number U653 and Gates IT Solution Sdn. Bhd. under its publication scheme.

References

[1] Europol's European Cybercrime Centre, "Police Ransomware Threat Assessment," no. February, 2014.
[2] A. Ajjan, "Ransomware: Next-Generation Fake Antivirus | SophosLabs Technical Paper," 2013. [Online]. Available: https://www.sophos.com/en-us/why-sophos/our-people/technical-papers/ransomware-next-generation-fake-antivirus.aspx.
[3] N. Andronio, S. Zanero, and F. Maggi, "HELDROID: Dissecting and detecting mobile ransomware," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9404, pp. 382–404, 2015.
[4] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Comput.*, pp. 1–15, 2014.
[5] S. Cesare and Y. Xiang, "Classification of malware using structured control flow," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 107, pp. 61–70, 2010.
[6] Q. Liao, "Ransomware : a Growing Threat To Smes How Ransomware Works ?," no. 2004, pp. 360–366.
[7] T. Rains, "Scareware : Don ' t Let Scammers Scare You," no. May, p. 2012, 2012.
[8] M. Schlesinger and V. Hlavác, "Supervised and unsupervised learning.," *Artif. Intell.*, no. April, 2011.
[9] A. S. Raza Ali, Usman Ghani, "Data Clustering and Its ApplicAtions," 2016. [Online]. Available: http://members.tripod.com/asim_saeed/paper.htm. [Accessed: 19-May-2016].
[10] Sophos, "Stopping Fake Antivirus: How to Keep Scareware Off Your Network," 2011.
[11] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, "DroidMat: Android malware detection through manifest and API calls tracing," *Proc. 2012 7th Asia Jt. Conf. Inf. Secur. AsiaJCIS 2012*, pp. 62–69, 2012.
[12] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-Based Malware Detection System for Android," *Proc. 1st ACM Work. Secur. Priv. smartphones Mob. devices - SPSM '11*, p. 15, 2011.
[13] Z. Aung and W. Zaw, "Permission-Based Android Malware Detection," *Int. J. Sci. Technol. Res.*, vol. 2, no. 3, pp. 228–234, 2013.
[14] D. D. Hosfelt, "Automated detection and classification of cryptographic algorithms in binary programs through machine learning," 2015.
[15] Y. Mishina, R. Murata, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, "Boosted random forest," *IEICE Trans. Inf. Syst.*, vol. E98D, no. 9, pp. 1630–1636, 2015.
[16] Dino Sejdinovic, "Statistical Data Mining and Machine Learning," no. 1998, 2006.
[17] R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, and D. S., "WEKA Manual for Version 3-6-13," 2015.
[18] Computer Crime and Intellectual Property Section (CCIPS). "How To Protect Your Network From Ransomware. [online]. Available: https://www.justice.gov/criminal-ccips/file/872771/download
[19] J. Crowe.," Ransomware Trends and Forecasts" 2017.[online]. Available: https://blog.barkly.com/new-ransomware-trends-2017
[20] K. Chen, "Algorithm On Clustering, Orientering and Conflict-Free Coloring," 2007.