IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 30 January 2023, accepted 8 March 2023, date of publication 16 March 2023, date of current version 22 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3258377

RESEARCH ARTICLE

Apk2Audio4AndMal: Audio Based Malware Family Detection Framework

OGUZ EMRE KURAL[1], ERDAL KILIÇ[1], AND CEYDA AKSAÇ[2]
[1]Department of Computer Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey
[2]Rönesans Holding, 06540 Ankara, Turkey

Corresponding author: Oguz Emre Kural (oguz.kural@bil.omu.edu.tr)

ABSTRACT Due to Android's popularity, cybercriminals view it as a lucrative target. Malwares with varying behavior patterns that specifically target user routines are constantly entering the market. Because of this, knowing how to identify different forms of malware is crucial for protecting against it. This paper proposes an audio-based malware family detection approach to achieve this goal. Android applications were converted to audio files in.wav format, and their audio-based features were extracted. Then, CFS-Subset, ReliefF, Information Gain, and Gain Ratio feature selection methods were applied to the extracted features. By examining the subsets obtained, features with high discrimination in Android malware family detection were determined. Classification experiments were conducted with the dataset created by randomly selected 500 samples from 8 families in AMD and Drebin datasets. Experiments with five different classifiers showed that effective malware family classification could be made with a small number of features in the audio domain.

INDEX TERMS Android, malware detection, family classification, audio based, feature selection, machine learning.

I. INTRODUCTION

The number of applications for mobile devices is expanding exponentially. People have access to several applications, including banking applications, social networking applications, health applications, and other options that hold personal information. Thus, mobile platforms become a direct target for malicious individuals. In addition to being open source, the Android operating system's diverse application ecosystems and dominant market share make it an attractive target. Although numerous steps are implemented against malicious software in application markets, these efforts are insufficient to protect end users from dangers. McAfee's 2021 mobile thread report indicates that more than 700,000 users downloaded it before a specific type of malware posted to the Google Play Store was found and removed [1].

According to the data provided by AV-Test, more than one million Android malware from different families were detected in 2022 [2]. Each kind of malware has unique objectives, and its tactics correspond to those objectives. Because of these differences in goals and approaches, family identification is more important to take the right actions.

In this paper, we propose a technique for detecting malware families using audio features. By transforming binary program files into audio files, we were able to extract audio-based features.

We employed several feature selection techniques to identify the family detection-effective features. In addition to similar studies, we conducted experiments with three additional features. Finally, to determine the malware's family, we applied five different classification algorithms during the classification process.

A. MOTIVATION

In recent years, Android malware detection and classification of Android malware according to their families is one of the important issues. Due to the importance of this issue, many new systems have been proposed and will continue to be proposed in recent years. Despite all these developments, some malware and families can easily bypass these detection systems [3]. Therefore, the need for up-to-date and different detection systems is increasing day by day.

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

Especially considering systems based on static analysis, these systems are mostly vulnerable to zero-day attacks [4]. Therefore, in addition to approaches such as static analysis, malware detection systems based on image and audio processing have been suggested by researchers in recent years [5]. In this study, a framework called Apk2Audio4AndMal based on audio processing, which is used less frequently than image and has become increasingly popular in recent years is proposed. With this framework, APK files can be considered as a more effective system against zero-day attacks, as a different representation system has emerged by converting to audio files. Because vectors with similar properties can often be seen in a static analysis or signature-based systems. However, since the structure will be completely different in audio files, the representation of each malware family or malware may also change. In addition to such advantages, there is no similar study that performs feature selection in audio-based Android malware family classification [6]. To the best of our knowledge, this will be the first study to examine feature selection and its effects on classification performance in audio-based Android malware family classification.

B. CONTRIBUTION

The main contributions of the study are as follows:

- Android applications were converted to audio files in.wav format and represented with audio-based features. Three features were added to the features used in addition to similar studies.
- Four feature selection methods were applied to see the effect of feature selection methods in audio-based Android malware family detection.
- Audio-based features were examined, and the features with high and low discrimination in Android malware family detection were determined.

C. RELATED WORKS

Mercaldo and Santone [7] proposed a two-stage malware detection and family classification method in their study in 2021. They extracted audio-based features by representing APK files as audio samples. They performed malware detection and family classification by training the two classification models with the extracted features. In the results they shared in terms of F-Measure, they reported the highest result for malware detection as 0.947 and the highest result for family classification as 0.914. They stated that they obtained the highest results with a 4-layer neural network.

Casolare et al. [8] performed Android malware family detection using audio-based features in 2021. They have extracted some numerical features from the audio file by converting the.dex (Dalvik executable) files in the apk files to audio. They performed classification processes by creating vectors representing each application with Chromagram, Root Mean Square, Spectral Centroid, Bandwidth, Spectral Rolloff, Zero Crossing Rate, Mel-Frequency Cepstral Coefficients, and Poly and Tonnetz properties. In their experiments with an unbalanced dataset containing 4746 samples from 10 families, the best result they obtained was 0.907 F-Measure, 0.988 accuracies.

Nataraj et al. [9] have proposed an orthogonal malware detection framework with audio descriptors, image similarity descriptors, and some static features. Research on using audio-based features in malware detection revealed the compatibility of audio-based features with other features. They used a metric based on the ‘‘Joint Feature Score’’ (JFS) to demonstrate the compatibility of feature sets with each other. Their results showed that the audio-based features were compatible with the image-based and static features.

In 2020, Zhang et al. [10] proposed a lightweight Android malware family detection method based on static attributes. They applied the classification steps by reducing the 8923 features obtained from the AndroidManifest.xml file with a simple feature selection approach. They reported an F1-score of 0.9851 with Logistic Regression in their experiments using an unbalanced 10-class data set. They stated that their method is fast and effective because the attributes are obtained only from the Manifest file. On feature selection steps, they select attributes based on features not used by more than one application. However, their feature selection method does not directly consider the discrimination of the selected features. In addition, their approach uses more than 4000 features even after the reduction process.

Fang et al. [11] proposed a malicious family detection method by fusing different features from.dex files. They expressed each application in a wide domain with the text, texture, and image-based attributes extracted from the.dex files. They performed the experiments with a total of 3000 samples using 15 families with more than 200 samples in the AMD dataset. Using a feature fusion algorithm based on multiple kernel learning for classification, they reported the best result as 0.96 F-Measure.

In [12], researchers visualized the characteristics of applications with different techniques. Using Control Flow Graph (CFG) and Data Flow Graph (DFG), encoded matrices were obtained for each application. They reported the best result obtained as 94.71% acc in the experiments they performed with the 20-class unbalanced dataset with the deep learning algorithm.

In their permission-based study conducted in 2018, Alswaina and Elleithy [13] performed feature selection with extremely randomized trees. They calculated the importance value in the range of 0-1 for each attribute by performing attribute selection on 59 attributes obtained from the dataset. Then, they eliminated the attributes with zero importance and reduced them to 42 attributes. They conducted experiments with six different classifiers on the dataset containing 1233 samples from 28 different families. In the results, they reported that they achieved a 95.99% accuracy with the RF algorithm.

D. ORGANIZATION

This study is organized as follows. In Section II, the flow of the proposed method is shared, and the application steps of the process are summarized. In Section III, the dataset used in the experimental studies and the performance metrics used in presenting the results are given. In Section IV, the results of the feature selection and classification stages are explained and presented in tables. Finally, in Section V, the results are discussed, and future study targets are given.

II. PROPOSED METHOD

This section explains the proposed method for malware family detection and the steps followed to create this method. The stages of the proposed method are shown in Figure 1.

The raw dataset consisting of apk files for malware family detection must be prepared for classification steps. To get the data ready, we ran a series of operations for each Android malware sample in the dataset. First, we exported the.dex files included in the apk files. Then, we read the.dex files in binary form, added the appropriate headers, and recorded them as audio files in.wav format. We have shared the flow of converting the.dex file to a.wav format audio file in Algorithm 1.

Since the dataset samples are represented as an audio file, it is now ready to extract audio-based features. We extracted the audio-based features for each audio sample and saved them to the CSV file with their family label. The extracted attributes are as follows. After the audio files are obtained from the.dex files, the attributes that can reveal the differences between the audio samples from each other are extracted. The following features were extracted from the audio samples.

- Chromagram is a representation of 12 pitch classes that captures the harmonic and melodic characteristics of sound [14], [15].
- RMS, which stands for root mean square, is a tool that measures the loudness of a sound sample within a window. The resulting value is an average of the total power of the audio sample.
- Spectral centroid indicates the center of mass of the spectrum. It is calculated by the weighted average of the frequencies present in the signal.
- Spectral Bandwidth, bandwidth is the difference between the highest and lowest values of frequencies in a sound sample.
- Spectral Rolloff is the frequency below which a specified percentage of the total spectral energy
- Zero Crossing Rate is a measure of how often the signal crosses zero per unit time. Speech discrimination is frequently used in audio applications such as music genre recognition. It is one of the simplest audio-based features [16].
- Spectral contrast is calculated by averaging the decibel difference between peaks and valleys in each frame in the spectrum. High contrast values indicate clear sound signals, while low contrast values indicate noisy sound signals [17].
- Flatness refers to how uniformly the frequencies in a spectrum are distributed. In other words, it shows how noisy the sound sample is. Flatness takes a value in the range of 0-1, and as it gets closer to 1, the sound becomes white [18].
- Melspectogram is the spectrogram in which the frequencies are converted to the mel scale. It represents the sound as a single channel image as it contains time and frequency information at the same time.
- Poly returns the coefficients necessary to fit an nth-order polynomial into the columns of the spectrogram at each frame.
- Tonnetz (tonal network) is a graphical representation of tonal centroid features. It is a useful tool for understanding the harmonic structure of tonal audio.
- MFCC are coefficients that represent sound similar to human perception. It usually consists of coefficients between 10-20. It is extensively used in speech and speaker recognition applications.

Algorithm 1 Apk to Audio Conversion

```
Input: $Application_1, \ldots, Application_N$
Output: .wav files
1:  Function $Convert2Audio(Application_1, \ldots, Application_N)$

2:  for $i = 1$ to $N$ do
3:      export.dex file from apk file for $Application[i]$
4:      read.dex file as binary for $Application[i]$
5:      add.wav file headers to data
6:      create.wav output file
7:      set.wav parameters (on the 7th, 8th, and 9th lines, the
        parameter of the.wav file is set)
8:      $number\_of\_channels \leftarrow 1$
9:      $sample\_width \leftarrow 1$
10:     $frame\_rate \leftarrow 32768$
11:     write data to file
12:     close file
13: end for
14: return .wav files
15: end Function
```

The feature vector was extracted for each file and saved with the tag indicating the malware family. After the feature extraction, a 4000×32 data matrix was obtained. In the resulting CSV file, each row represents a sample from the dataset, and each column represents an attribute extracted from the samples. At this stage, the CSV file can be used to train machine-learning algorithms for malware family detection. However, it is desired to investigate whether all extracted features effectively discriminate malware families. For this reason, feature selection was made on the obtained CSV file with CFS-Subset, Information Gain, Gain Ratio, and ReliefF algorithms. After the selections were made, the data were reduced according to the selected features, and reduced data sets were obtained. Details of the feature selection methods used are given in Section II-A. To make family classification, creating a model in classification algorithms is necessary using the data obtained in CSV files. For this process,

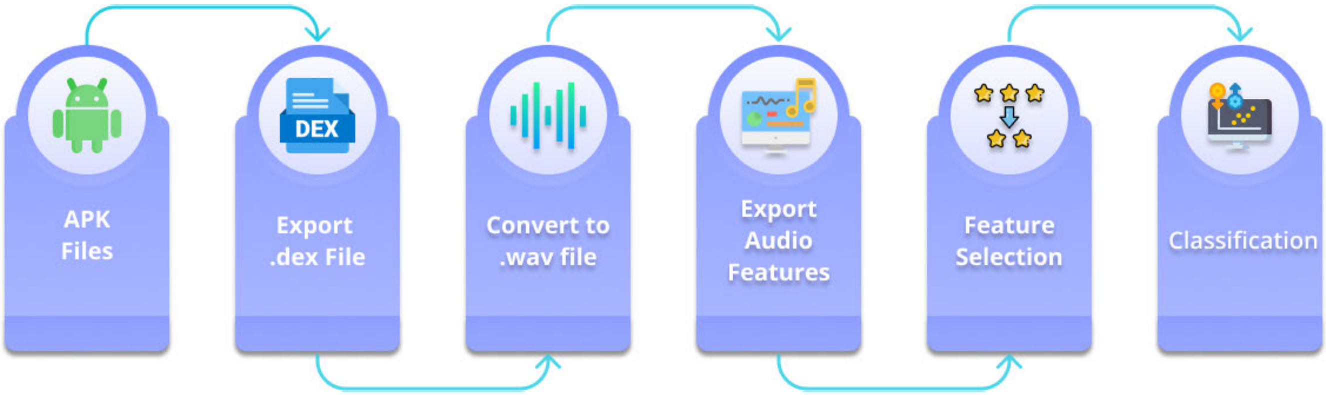


FIGURE 1. Audio features based android malware family detection workflow.

classification experiments were performed with KNN, SVM, Logistic, Random Forest, and C4.5 algorithms.

A. FEATURE SELECTION

Suppose *FS* is a set showing all attributes of a dataset. Finding the best subset that can be selected from this set is called feature selection. The goodness of the selected subset is the situation in which the selection is made in a way that does not adversely affect the classification performance. There are many advantages to using feature selection methods. These can be considered as reducing the computational cost, eliminating the excessive memorization problem, and running machine learning techniques efficiently. Feature selection methods are generally evaluated under 3 groups [19]. These are filter-based feature selection methods, wrapper feature selection methods, and embedded feature selection methods. The feature selection methods used in this study are discussed in detail in the subsections.

1) INFORMATION GAIN (IG)

An attribute's "information gain" (IG) indicates how much data it provides about a given class. The information gain metric employs entropy from the theory of information. In practice, it is calculated based on the difference in entropy before and after the data is separated by an attribute. Equation 1 provides a purely mathematical formulation of the IG metric. For feature f, the IG score is found using Equation 1.

$$IG(f, D) = Entropy(D) - \left(\sum_{j=1}^{n} \frac{|D_j|}{|D|} * Entropy(D_j) \right) \quad (1)$$

In Equation 1, D represents the used dataset, f represents the evaluated feature, and $|Dj|$ represents the number of times the j value passes in f.

2) GAIN RATIO (GR)

The IG method becomes biased when the number of distinct values a feature has is large. This is because the number of branches after division is high, and the number of samples under each branch is low. This raises problems such as overfitting. To prevent this situation, the Gain Ratio algorithm normalizes the Information Gain algorithm with SplitInfo and detects features with high representation ability. The Gain Ratio algorithm is shown in Equation 2.

$$Gain - Ratio = \frac{IG(f, D)}{SplitInfo} \quad (2)$$

3) CFS SUBSET (CSF)

Correlation-Based Feature Selection (CFS Subset) [20] is a feature selection algorithm that aims to select the best feature set by calculating the correlation between features. The algorithm aims to select a subset from the feature set that has high representativeness (low correlation between them) and high correlation with the class label. The correlation between features is calculated according to Equation 3.

$$Cor(X, Y) = \frac{(N * \sum(X_i * Y_i) - (\sum X_i) * (\sum Y_i))}{std(X) * std(Y))} \quad (3)$$

The numerator part of the formula expresses the covariance between X and Y features, and the denominator part expresses the product of the standard deviations of the X and Y attributes.

4) ReliefF (RFF)

Relief is a filter-based feature selection algorithm proposed by Kira and Rendell [21]. Although it was a simple and effective method, it could only deal with two-class problems. After developing several intermediate versions, Kononenko 1994 proposed the ReliefF algorithm [22], the sixth version of the Relief algorithm (A, B,.., F), which can be applied to multi-class problems. While the ReliefF algorithm works on a multi-class dataset, R samples are selected from the dataset in each m iteration. Then, k Nearest Hits from the same class and k Nearest Misses for each different class are found for the selected sample. The weights of the attributes are updated according to the values found. As a result of iterations, the updated weights in each round take the result values. The higher the resulting weight values, the more valuable the features are considered for classification.

B. CLASSIFICATION

Machine learning approaches generally focus on classification, clustering, and regression problems [23]. While classification and regression are accepted as supervised learning approaches, clustering is one of the unsupervised learning approaches. While labels or values are learned in the training phase of supervised learning techniques, labels are not learned in clustering. In addition to supervised and unsupervised learning, there are also semi-supervised learning and reinforcement learning structures. In this study, the classification problem is handled. Because malware is separated into families.

WEKA is a tool that contains many machine learning and data mining algorithms [24]. In this study, malware is separated according to families by using Random Forest (RF), K-Nearest Neighbors (KNN), C4.5, Logistic Regression (LR), and Support Vector Machine (SVM) algorithms included in the WEKA tool. Classification results are calculated by using predefined values in the parameters of these algorithms. The infrastructures of these algorithms are summarized as follows:

1) RANDOM FOREST (RF)

Decision trees analyze the classes of the training dataset. Using this information, the inference is made to which class the test data belong. By creating many if, else-if, else rules, it is decided to which class the relevant data will belong. In the random forest algorithm, a large number of decision tree algorithms are brought together, and this algorithm makes a decision. This is why the word "forest" is used in the name of the algorithm. It is an algorithm based on community learning. The main idea behind this classifier is that each tree votes for a class, and the forest generates a large number of unbiased decision trees from random samples of the training data, with the replacement method choosing the classification with the most votes among all the trees in the forest. The Gini index is used to construct decision trees and determine the last class in each tree.

2) K-NEAREST NEIGHBORS (KNN)

It is a classification algorithm created by looking at the distances between the attributes of the data. In the infrastructure of this algorithm, metrics such as Euclidean distance and Minkowski distance are used. By means of these metrics, the distances between the data are measured. For example, suppose we have a dataset consisting of two attributes. Let these attributes be x and y. The distance between data A and data B can be calculated as $\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ according to the Euclidean distance. The k parameter in this algorithm is selected by the user. Class estimation is performed by looking at the classes of the k samples at the least distance from the data whose class label is unknown. For example, let's assume that the value of k is selected as 3. The data with an unknown label is labeled by looking at the class of the 3 data closest to a sample with an unknown label. The main idea here is which class is most present.

TABLE 1. Used dataset.

Family	Count	Origin Dataset
Bankbot	500	AMD
DroidKubgFu	500	AMD
FakeInst	500	AMD
Fusob	500	AMD
Jisut	500	AMD
Mecor	500	AMD
Opfake	500	Drebin
Plankton	500	Drebin

3) C4.5 DECISION TREE

It is one of the widely used classification algorithms in the field of machine learning. Decision trees are very easy to interpret and understand [25]. With this structure, data is processed quickly. The most important step in the infrastructure of decision trees is to determine the branch of the tree according to which criteria. By eliminating this problem, it is determined according to which property values the decision tree structure will be created. C4.5 classification learning algorithm is a decision tree algorithm proposed by Quinlan [26]. This algorithm is very efficient and is one of the most frequently used decision trees. It has more diverse and newer learning algorithms than the ID3 algorithm. This algorithm has emerged as the current version of the ID3 algorithm. The C4.5 algorithm uses the gain ratio metric to solve the limitation that the ID3 algorithm is highly sensitive to multi-valued features.

4) LOGISTIC REGRESSION (LR)

In some datasets, the variables consist of binary structures such as positive-negative, successful-unsuccessful, yes-no, and satisfied-not satisfied. As is the case here, if the dependent variable consists of two-level or multi-level categorical data; Examining the cause-effect relationship between the dependent variable and the independent variable can be calculated quite easily with Logistic Regression Analysis. In addition, the classification process is carried out with logistic regression analysis. In this technique, there is no normal distribution and continuity assumption prerequisite. The effects of explanatory variables on the dependent variable are obtained as probabilities, and the risk factors are determined as probabilities. It is mainly used in solving binary classification problems. However, it can also be used in solving classification problems with more than two numbers with the one-versus-all approach.

5) SUPPORT VECTOR MACHINE (SVM)

It is a classification algorithm based on the linear separability principle proposed by Cortes and Vapnik [27]. This algorithm is a technique that sees the input data in an N-dimensional space as two sets of vectors. In order to make a good classification, it is aimed to maximize the bound-

TABLE 2. Feature selection techniques and selected features.

	ReliefF	Gain-Ratio	CFS-Subset	Infogain
chroma_stft	x	x	x	x
rms	x	x	x	x
spectral_centroid		x		x
spectral_bandwidth				x
rolloff		x		x
zero_crossing_rate				
contrast	x	x		x
flatness	x	x		x
melspectrogram	x	x	x	x
poly0	x	x	x	x
tonnetz	x			
mfcc1	x	x	x	x
mfcc2	x			
mfcc3	x	x		x
mfcc4	x	x	x	
mfcc5	x			
mfcc6				
mfcc7		x		x
mfcc8				
mfcc9		x		
mfcc10	x	x		x
mfcc11				
mfcc12				
mfcc13				
mfcc14	x			
mfcc15				
mfcc16				
mfcc17				x
mfcc18	x	x		x
mfcc19				
mfcc20	x			

ary of the hyperplane to be passed between the two classes to be classified. To find the appropriate hyperplane, two parallel lines are passed over the closest members of the two classes of the data set, and the line in the middle of these two lines is called the hyperplane. Parallel lines are the boundaries of the hyperplane, while support vectors are vectors that pass over the nearest members of the datasets. The biggest advantage of the Support Vector Machine (SVM) algorithm is that in case the problem cannot be solved, the relevant space is changed, and the solution to the problem is searched again. In this way, a high-performance classification algorithm is built by providing a better separation. There is no direct SVM algorithm in the WEKA package. However, there is an improved Sequential Minimal Optimization (SMO) algorithm to solve the optimization problem more efficiently [28].

III. EXPERIMENTAL SETTINGS

This section will explain the data set, feature extraction, feature selection, and classification steps used in the experiments.

A. DATASET

We performed our experiments on a balanced 8-class dataset obtained from different datasets. We created a homogeneous dataset consisting of 4000 data in total by choosing eight classes with more than 500 samples from AMD and Drebin [29] datasets. The malware families used in the experiments and which datasets they were taken from are shown in Table 1.

B. PERFORMANCE MEASURES

To accurately express the results of the experiments, the results must be presented with the right metrics. For this reason, the results were interpreted by the Precision, Recall, and F-Measure metrics. Precision and Recall metrics are where the positive class is at the forefront. Precision refers to how much of the positively predicted data is genuinely positive. Recall, on the other hand, expresses how much of the truly positive data is predicted as positive. The equations of Precision and Recall are given in Equation 4 and Equation 5.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

TP refers to those in the positive class and correctly classified, and FP refers to those who are not in the positive class but are classified as positive. FN refers to those who are classified as negative but are positive.

F-Measure calculation using Precision and Recall values:

$$2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

IV. RESULTS AND DISCUSSIONS

The results obtained by applying four different feature selection methods on the dataset are given in Table 2. When the table is examined, it is seen that six features are selected with the CFS-Subset method, 15 features are selected with the Gain Ratio method, 15 features are selected with the Information Gain method, and 16 features are selected with the ReliefF method on total 31 features. It has been observed that all methods select three common features (chroma-stft, rmse, melspectrogram), and three or more methods select nine common features. In the experiments, it was observed that all methods selected the Mel spectrogram we added to the feature set, and all three methods except CFS-Subset selected the flatness and contrast. The high selection rate of the added features indicates that their discrimination on the dataset is high.

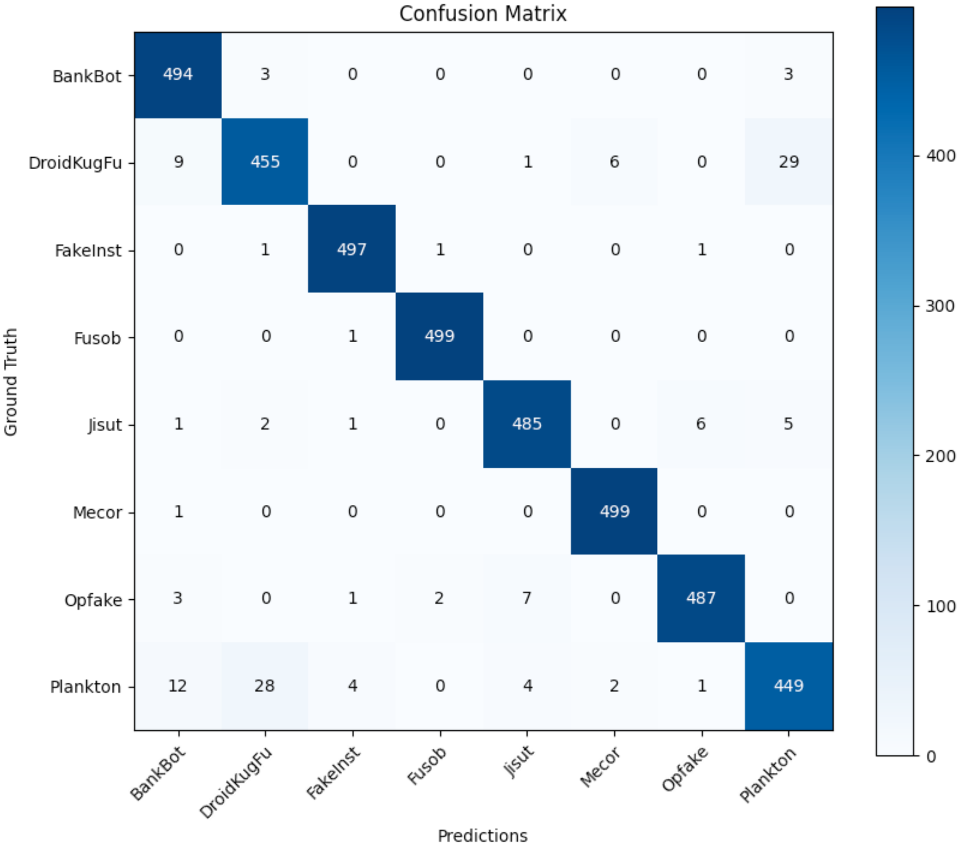
When the low number of selected features is evaluated, it is seen that zero crossing rate, mfcc6, mfcc8, mfcc11, mfcc12, mfcc13, mfcc15, mfcc16, mfcc19 features are not selected by any algorithm. Tonnetz, mfcc5, mfcc14, and mfcc20 attributes are selected only by the ReliefF algorithm.

In family classification experiments, it was preferred to use Weka, which has many classifiers ready. Classification results were obtained for each data set reduced by feature selection algorithms with KNN, Random Forest, C4.5, Logistic, and SVM algorithms. To better understand the generalizability of the models created with the classifiers, 10-fold cross-validation was applied to all classifiers. The results obtained with the classifiers according to the feature selection methods are shown in Table 3. The findings observed in the experiments are given below.

The results obtained with KNN and Random Forest are generally very close. The highest performance in all

TABLE 3. Classification results.

	CFS-Subset (6)	Gain-Ratio (15)	Infogain (15)	ReliefF (16)	All (31)
Random Forest	0.952	0.961	0.961	0.962	0.961
KNN	0.94	0.961	0.961	**0.966**	**0.962**
Logistic	0.671	0.823	0.834	0.832	0.89
C4.5	0.912	0.931	0.931	0.936	0.934
SMO	0.601	0.707	0.708	0.768	0.783



	BankBot	DroidKugFu	FakeInst	Fusob	Jisut	Mecor	Opfake	Plankton
BankBot	494	3	0	0	0	0	0	3
DroidKugFu	9	455	0	0	1	6	0	29
FakeInst	0	1	497	1	0	0	1	0
Fusob	0	0	1	499	0	0	0	0
Jisut	1	2	1	0	485	0	6	5
Mecor	1	0	0	0	0	499	0	0
Opfake	3	0	1	2	7	0	487	0
Plankton	12	28	4	0	4	2	1	449

FIGURE 2. Audio features based android malware detection workflow.

experiments was obtained using the ReliefF feature selection method and KNN with 0.966. The lowest-performing classifier for all feature selection methods is SVM. SVM had the highest success with 0.783 on the non-reduced dataset. However, even this value is relatively low compared to other classifiers. In classification experiments with data reduced by CFS-Subset, the best result was obtained with Random Forest with a score of 0.952. The lowest results were obtained with SMO and Logistic, respectively. Although classification was made using only six features, quite acceptable classification performances were obtained with RF and KNN. The results obtained in classification experiments with data reduced by Gain-Ratio and Information gain are very close. Among both feature selection methods, the best classification results with 0.961 were obtained with KNN and Random Forest algorithms. Experiments with data reduced by ReliefF generally gave high results for all classifiers. RF, KNN, and C4.5 algorithms achieved higher results with the features selected with ReliefF compared to the classifications made using the whole data set.

The confusion matrix for the results obtained using ReliefF and KNN is shown in Figure 2. In Table 4, the performance metrics of the same configuration are shown.

When Table 4 is examined, it is seen that the Fusob and Mecor families have the best classification results and the lowest error rate, with one misclassification each. Only one sample of Fusob is classified as FakeInst. Similarly, a sample from the Mecor family is classified as BankBot. However, since the number of other classes classified as Fusob is less than those classified as Mecor, the precision of Fusob is higher than Mecor. One instance of FakeInst and two instances of Opfake from other families are classified as Fusob.

TABLE 4. Performance metrics using ReliefF and KNN.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area
BankBot	0.988	0.007	0.950	0.988	0.969	0.964	0.995
DroidKugFu	0.910	0.010	0.930	0.910	0.920	0,909	0,976
FakeInst	0.994	0.002	0.986	0.994	0.990	0.989	0.996
Fusob	0.998	0.001	0.994	0.998	0.996	0.995	0.999
Jisut	0.970	0.003	0.976	0.970	0.973	0.969	0.991
Mecor	0.998	0.002	0.984	0.998	0.991	0.990	0.998
Opfake	0.974	0.002	0.984	0.974	0.979	0.976	0.992
Plankton	0.898	0.011	0.924	0.898	0.911	0.898	0.970
Average	0.966	0.005	0.966	0.966	0.966	0.961	0.990

TABLE 5. Comparison of previous works.

References	Year	Type	MD	FC	NC	NF	Precision	Recall	Result
[7]	2021	Audio	Yes	Yes	71	28	0.911	0.913	0.922 Acc
[8]	2021	Audio	No	Yes	10	29	0.907	0.907	0.988 Acc 0.907 F-Measure
[10]	2020	Manifest Features	No	Yes	10	3712	-	-	0.983 F-Measure
[11]	2020	Image Text Texture	No	Yes	15	64+	0.960	0.960	0.960 F-Measure
[12]	2019	CFG and DFG Virtualization	No	Yes	20	313+	-	-	0.947 Acc
[13]	2018	Permissions	No	Yes	28	42	-	-	0.959 Acc
[30]	2020	Hybrid Features	No	Yes	21	329	-	-	0.9804 Acc
Our	**2023**	**Audio**	**No**	**Yes**	**8**	**16**	**0.966**	**0.966**	**0.966 F-Measure**

MD: Malware Detection, **FC:** Family Classification, **NC:** Number of Class, **NF:** Number of Features, **Acc:** Accuracy

Table 5 compares some studies and the results obtained from this study. The survey on Android malware family detection did not consider an audio-based approach [6]. As far as we examined, only two studies were found [7], [8]. When the results of both studies were examined, it was observed that the audio-based methods gave good results in malware family detection. Likewise, the results obtained from this study are remarkable. In addition, as far as we have examined, there has been no study that makes feature reduction in audio-based studies in the domain we are working on.

When the result of this study is compared with other studies, it was seen that good results were obtained with very few features. For example, while 3712 features were used in the study [10], close results were obtained with only 16 features in this study. Also, an utterly balanced dataset using only six features with the CFS Subset method yielded more than 95% performance in this study. In the studies of the classification of Android malware according to their families, it has been seen that primarily unbalanced datasets are used, whereas a metric such as Acc, which may be a problem in comparison, is preferred instead of a metric such as F-measure. In this study, a completely balanced and up-to-date dataset is handled, and results are given with the F-measure, which is fair to compare.

V. CONCLUSION AND FEATURE WORKS

In this study, we moved away from classical approaches and evaluated the application files in the audio domain and benefited from the features of this domain. In other words, we performed malware family detection over application files that we converted to audio. We applied feature selection algorithms to detect features with high discrimination in malware family detection by adding 3 features in addition to 28 audio-based features included in similar studies. In our evaluations, we saw that all methods selected one of the three features we added in addition to similar studies. The other two features were selected by three methods each. This shows that the added features have high discrimination in malware family detection. In addition, in our experiments with 4000 samples with eight classes, we revealed that high classification success could be achieved with only half of the extracted features. In future studies, we aim to conduct research for a hybrid malware and malware family detection method by hybridizing the previously proposed image-based approach [31] and the audio-based approach proposed in this study.

REFERENCES

[1] (2021). *McAfee Mobile Threat Report*. [Online]. Available: https://www.mcafee.com/content/dam/global/infographics/McAfeeMobileThreatReport2021.pdf

[2] (2022). *Malware Statistics Trends Report*. [Online]. Available: https://www.av-test.org/en/statistics/malware/

[3] V. Kouliaridis, K. Barmpatsalou, G. Kambourakis, and S. Chen, "A survey on mobile malware detection techniques," *IEICE Trans. Inf. Syst.*, vol. 103, no. 2, pp. 204–211, 2020.

[4] P. Yan and Z. Yan, "A survey on dynamic mobile malware detection," *Softw. Qual. J.*, vol. 26, no. 3, pp. 891–919, 2018.

[5] C. Bijitha and H. V. Nath, "On the effectiveness of image processing based malware detection techniques," *Cybern. Syst.*, vol. 53, no. 7, pp. 615–640, 2022.

[6] F. Alswaina and K. Elleithy, "Android malware family classification and analysis: Current status and future directions," *Electronics*, vol. 9, no. 6, p. 942, Jun. 2020.

[7] F. Mercaldo and A. Santone, "Audio signal processing for Android malware detection and family identification," *J. Comput. Virol. Hacking Techn.*, vol. 17, no. 2, pp. 139–152, Jun. 2021.

[8] R. Casolare, G. Iadarola, F. Martinelli, F. Mercaldo, and A. Santone, "Mobile family detection through audio signals classification," in *Proc. 18th Int. Conf. Secur. Cryptogr. (SECRYPT)*. Setúbal, Portugal: SciTePress, 2021, pp. 479–486.

[9] L. Nataraj, T. M. Mohammed, T. Nanjundaswamy, S. Chikkagoudar, S. Chandrasekaran, and B. S. Manjunath, "OMD: Orthogonal malware detection using audio, image, and static features," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 703–708.

[10] Y. Zhang, C. Feng, L. Huang, C. Ye, and L. Weng, "Detection of Android malicious family based on manifest information," in *Proc. 15th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2020, pp. 202–205.

[11] Y. Fang, Y. Gao, F. Jing, and L. Zhang, "Android malware familial classification based on DEX file section features," *IEEE Access*, vol. 8, pp. 10614–10627, 2020.

[12] Z. Xu, K. Ren, and F. Song, "Android malware family classification and characterization using CFG and DFG," in *Proc. Int. Symp. Theor. Aspects Softw. Eng. (TASE)*, Jul. 2019, pp. 49–56.

[13] F. Alswaina and K. Elleithy, "Android malware permission-based multi-class classification using extremely randomized trees," *IEEE Access*, vol. 6, pp. 76217–76227, 2018.

[14] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 96–104, Feb. 2005.

[15] A. Shah, M. Kattel, A. Nepal, and D. Shrestha, "Chroma feature extraction," pp. 1–13, 2019. [Online]. Available: https://www.researchgate.net/profile/Ayush-Shah-6/publication/363487456_Chroma_Feature_Extractionpdf/data/631f9a1770cc936cd301efc1/Chroma-Feature-Extraction.pdf

[16] T. Giannakopoulos and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB Approach*. New York, NY, USA: Academic, 2014.

[17] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 1, Aug. 2002, pp. 113–116.

[18] S. Dubnov, "Generalization of spectral flatness measure for non-Gaussian linear processes," *IEEE Signal Process. Lett.*, vol. 11, no. 8, pp. 698–701, Aug. 2004.

[19] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[20] M. A. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1999.

[21] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. AAAI*, vol. 2, 1992, pp. 129–134.

[22] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Machine Learning: ECML-94*, F. Bergadano and L. De Raedt, Eds. Berlin, Germany: Springer, 1994, pp. 171–182.

[23] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, pp. 1–21, May 2021.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, Jun. 2009.

[25] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[26] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Mar. 1996.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[28] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft, Tech. Rep. MSR-TR-98-14, Apr. 1998. [Online]. Available: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/

[29] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "DREBIN: Effective and explainable detection of Android malware in your pocket," in *Proc. NDSS*, vol. 14, 2014, pp. 23–26.

[30] O. F. T. Cavli and S. Sen, "Familial classification of Android malware using hybrid analysis," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ISCTURKEY)*, Dec. 2020, pp. 62–67.

[31] O. E. Kural, D. O. Şahin, S. Akleylek, E. Kılıç, and M. Ömüral, "Apk2Img4AndMal: Android malware detection framework based on convolutional neural network," in *Proc. 6th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2021, pp. 731–734.

OGUZ EMRE KURAL received the bachelor's degree in computer engineering from Karadeniz Technical University, Trabzon, in 2013, and the master's degree in computer engineering from Ondokuz Mayıs University, Samsun, in 2017, where he is currently pursuing the Ph.D. degree in computational sciences. His research interests include machine learning, image processing, data mining, and Android malware analysis.

ERDAL KILIÇ received the bachelor's and master's degrees in electrical and electronic engineering from Karadeniz Technical University, Trabzon, in 1991 and 1996, respectively, and the Ph.D. degree in electrical and electronic engineering from Middle East Technical University, Ankara, in 2005. He is currently a Full Professor with the Department of Computer Engineering, Ondokuz Mayıs University. His research interests include neural networks, machine learning, and data mining.

CEYDA AKSAÇ received the bachelor's degree in computer engineering from Başkent University, in 2011, where she is currently pursuing the master's degree. She has been with Rönesans Holding, since 2013. Her research interests include machine learning, data mining, and neural networks.