SEMARAK ILMU PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Intersection Filtering based on Recursive Feature Elimination Cross-Validation to Improve Classification Models in Early Detection of Android Malware

Akhmad Dahlan[1,*], Yoga Pristyanto[2], Muhammad Shahkhir Mozamir[3], Muhamad Hariz Muhamad Adnan[4]

[1] Department of Informatics Management, Faculty of Computer Science, Universitas Amikom Yogyakarta, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia
[2] Department of Information System, Faculty of Computer Science, Universitas Amikom Yogyakarta, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia
[3] Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia
[4] Faculty of Computing and Meta-Teknologi, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia

ABSTRACT

Android malware is malicious software designed to damage or steal data on Android operating system devices. Machine learning models can be a solution for the early detection of Android malware. The problem in machine learning is that the large dimensions of the malware dataset can cause the model performance to be less than optimal. In this research, the proposed method of Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) is used for the process, which is expected to create a model that is robust to several types of high dimensional data, especially Android malware detection datasets. The research results show that Intersection Filtering based on RFECV (IF-RFECV) can produce fewer features and correlate with the label or target class. Overall, feature reduction using Intersection Filtering based on Recursive Feature Elimination Cross Validation (IF-RFECV) can produce accuracy, precision, recall and f-1 scores on the classification model that are better than original features or RFECV alone. The processing time carried out by Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) is similar to original features or RFECV alone. With the increase in results, this model can be used well in detecting malware on the Android operating system.

Keywords:
Features reduction; Machine learning; Classification; IFRFECV; Subset-features

1. Introduction

Android malware is malicious software designed to damage or steal data on An-droid operating system devices [1]. Android malware includes various types of malicious software, such as viruses, Trojans, worms and spyware, which can cause various problems on Android devices. This malware can be distributed through applications down-loaded from untrusted sources, malicious websites, or

[*] *Corresponding author.*
E-mail address: alland@amikom.ac.id

https://doi.org/10.37934/araset.61.1.127141

phishing messages [2,3]. The main goal of Android malware is to steal personal information, damage the device, or even control the device remotely [4]. Therefore, early treatment is needed to anticipate the presence of malware on the Android operating system. One approach that can be used is artificial intelligence (AI) technology.

The application of artificial intelligence (AI) technology in detecting and anticipating malware on Android devices has brought significant changes in the cybersecurity land-scape [5]. By leveraging machine learning, malware detection systems can dynamically analyse suspicious behavioural patterns, identify new variants and predict potential threats before exploiting system vulnerabilities. Sophisticated heuristic analysis, powered by artificial intelligence, allows the system to differentiate between regular activity and signs of unconventional malware attacks, such as attempts to trick detection [4]. Additionally, deep learning in malware prevention allows systems to update and improve themselves automatically, providing a more adaptive response to evolving threats [6,7].

Artificial intelligence technology also enhances anomaly detection capabilities, al-lowing the system to recognize behaviour that does not conform to standard usage patterns. Automatic updates and AI adaptability help maintain system resilience against increasingly complex malware evolution [8]. By integrating AI technology into Android malware detection and anticipating, cybersecurity companies can provide more effective and responsive protection, making the Android user experience safer and more secure from various cyber threats that can be detrimental [9].

One AI approach that can be used is Machine Learning. Machine learning models can be programmed to understand application behaviour patterns; such as access to sensitive data or suspicious changes in system configuration. By comparing application activity with learned normal behaviour, the model can detect potential malware based on identified anomalies [7,10]. Although machine learning techniques have successfully improved the effectiveness of Android malware detection, growing challenges include the speed of malware adaptation, variations in attack techniques and efforts to minimize false positives [11,12].

Several studies have been carried out regarding the application of machine learning to detect Android malware. Research conducted by Burak *et al.,* [13] used the Support Vector Machine algorithm to classify the presence of Android malware. The results of this re-search are adequate, with an accuracy of more than 94%. However, the research should have stated the number of features used in the dataset. Similar research was also con-ducted by Fiky *et al.,* [14], who used the Drebin+Malgenome dataset with 215 features. In this study, feature selection was done using a combination of Information Gain (IG) and Principal Component Analysis (PCA) to reduce the number of features in the dataset. As a result, the remaining features selected were 24 with a classifier accuracy value of 98%. However, this research must state how long the computing process takes expressly. On the other hand, research conducted by Islam Rejwana *et al.,* [15] uses Principal Component Analysis (PCA) to reduce the number of features in the dataset. The dataset, which initially had 141 features, was reduced to 45. This research resulted in the highest classification algorithm performance of 95%. In addition, research conducted by Mustaqim *et al.,* [16] uses Recursive Feature Elimination Cross Validation (RFECV) to perform feature reduction on the dataset. The results of this research show that feature reduction using RFECV can improve the performance of the classification model used. In addition, the number of features processed by the model is significantly reduced so that processing time can be faster.

Based on previous research, each technique only improves the performance of the classification model on specific datasets and cannot continuously improve the performance of the resulting classification model. In this research, a combination of filtering and wrapping techniques is proposed as a technique for carrying out the feature selection process. The proposed technique is an Intersection Filtering Model based on Recursive Feature Elimination Cross-Validation. The goal is to improve the performance of classification models for several types of datasets. By using the Intersection Filtering Model technique based on Recursive Feature Elimination Cross Validation to carry out the feature reduction process, it is hoped that we can create a model that is robust to several types of high dimensional data, one of which is the Android malware detection dataset.

There are three main contributions to this research: First, the proposed method (IF-RFECV) can produce a better model performance than previous research, such as single RFECV, PCA and Information Gain, especially in Android malware detection. The second, IF-RFECV methods can solve the problem of selecting relevant features in the dataset, especially in Android malware detection. The last is the IF-RFECV method can be a reference for further research on high-dimensional dataset problems and feature selection, especially in Android malware detection.

2. Methodology

This research uses several materials and tools to execute the proposed method. The tools used in this research consist of hardware and software. The following are the devices used—personal Computer in the form of a desktop computer for research. The computer specifications are that it has an AMD Rayzen 5 5600G processor. The AMD Ryzen™ 5 5600G processor has 6 CPU cores, a base clock speed of 3.9 GHz and 7 GPU cores. The computer memory used is 16GB DDR4 3200Mhz.

Meanwhile, the GPU used is NVIDIA RTX 2060 with 6GB GDDR6 operating at a clock speed of 1750 Mhz (effective data rate 14Gbps). The RTX 2060 is also equipped with 240 Tensor Cores and 30 RT Cores. The software used in this research includes Jupyter Notebook, Python 3.10 and Google Collaboratory. All three are used to implement the methods used in this research. They are also used to create models and test the performance of the models used. Meanwhile, the material used in this research is the Android Malware Classification datasets, which are publicly available on the UCI Repository and Kaggle Repository. Details of the dataset will be discussed in the data acquisition section.

This section will also discuss the flow of research carried out, including the proposed method used in this research. Figure 1 shows the research flow, including data acquisition, data preparation and feature reduction using the Intersection Filtering Model based on the Recursive Feature Elimination Cross Validation, Classification Model and Model Evaluation.

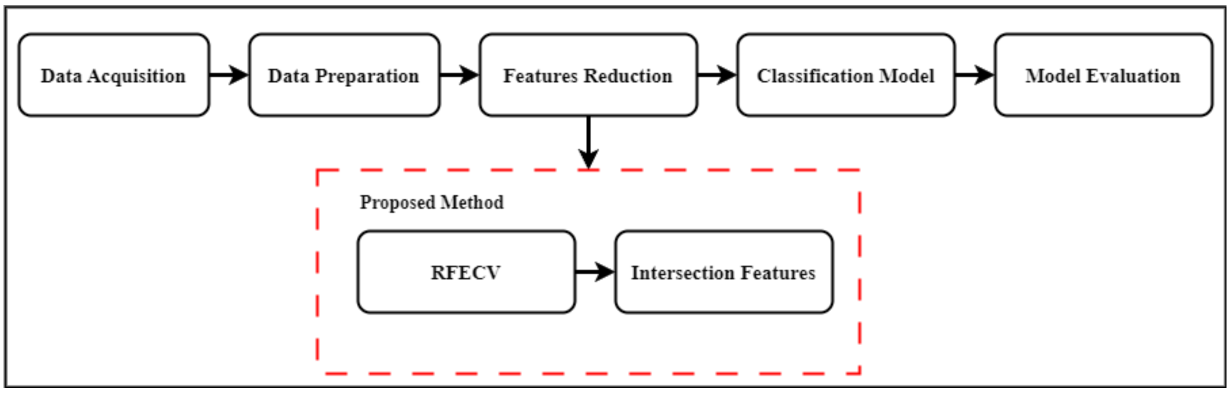

Fig. 1. Research workflow

Based on Figure 1 above, the following discussion of each step follows the research flow.

2.1 Data Acquisition

Data collection is a crucial initial stage because the dataset is the primary material used in this research. The dataset used in this study is a public dataset obtained from the UCI Machine Learning Datasets Repository. The dataset used is TUNADROMD. This dataset contains 4465 instances and 241 attributes. The target attribute for classification is a category. The class attribute is malware and good-ware [17,18]. Table 1 below is a sample dataset used in this research.

Table 1
Samples of the dataset used

ACCESS_ALL_DOWNLOADS	ACCESS_CACHE_FILESYSTEM	……..	Label
0	1	……..	Malware
1	1	……..	Malware
1	1	……..	Malware
1	0	……..	Malware
…	…	……..	…
0	0	……..	Good-ware

The dataset consists of two Android components, where feature numbers 1-214 are permission-based feature components on the Android operating system. Meanwhile, feature numbers 215-241 are components of API-based features on the Android operating system. Each feature has a binary value, namely 0 for "No" and 1 for "Yes". The class or target is a type of categorical value with two classes or labels for this dataset or binary classes, namely malware and good-ware. This TUNADROMD dataset has no missing, inconsistent, or duplicate values.

2.2 Data Preparation

In the data preparation stage, data cleaning is carried out. The cleaning process includes:

i. Filling in blank data
ii. Eliminating data duplication
iii. Checking for data inconsistencies
iv. Correcting errors in data.

Usually, missing values is caused by new data for which no information exists. However, no data duplication and missing values were found in the datasets used.

2.3 Features Reduction

At the feature selection stage, two approaches are used with the following steps:

i. The first stage uses the RFECV wrapping technique. The RFECV configuration used is the Decision Tree estimator, while for the base model, several popular algorithms are used, such as Gradient Boosting (GB), K-Nearest Neighbour (KNN), Decision Tree C4.5 (DT C4.5), Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP).
ii. RFECV produces a subset of new features from each base model used.

iii. The next stage is to filter the features using Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV). It looks for intersections of subsets of features produced from several RFECV-based base models in each dataset.
iv. The resulting features from Intersection Filtering based on Recursive Feature Elimination Cross Validation (IF-RFECV) will be used for the classification modelling process and model testing.

2.4 Classification Model

In this research, the classification model used is a popular model used by researchers in machine learning. Two types of classification models are used, namely single classifier and ensemble classifier models. The single classifier models used include K-Nearest Neighbour (KNN), Decision Tree C4.5 (DT C4.5), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Meanwhile, the ensemble classifier models used are Gradient Boosting (GB) and Random Forest (RF).

2.5 Model Evaluation

The final stage of this research is to evaluate the resulting model. Apart from that, at this stage, a comparison will also be made between the proposed Intersection Filtering method based on Recursive Feature Elimination cross-validation (IF-RFECV) and previous research models [16,19-21] to find out how well the resulting model performs. Several evaluation indicators are used in testing and comparing models, especially for classification models. The indicators used include Accuracy, Precision, Recall, F1 Score and Time Processing (seconds). Following is formulas Eq. (1), Eq. (2), Eq. (3) and Eq. (4) to calculate the values of these four indicators.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TN}{TP+FN} \tag{3}$$

$$F1\,Score = \frac{2\,x\,Precision\,x\,recall}{precision\,+recall} \tag{4}$$

The following is a description of each notation contained in the equation above. True Positive (TP) is the number of positive records classified as positive. False positives (FP) are the number of negative records classified as positive. False negatives (FN) are the number of positive records classified as negative. True negatives (TN) are the number of negative records classified as negative.

3. Results and Discussion

The results and discussion section focuses on the feature reduction process, classification model testing and comparative evaluation of classification models.

3.1 Features Reduction

The TUNADROMD dataset contains 241 attributes. Here are all the features in the dataset in Table 2.

Table 2
All features in TUNADROMD dataset

All Features Name
ACCESS_ALL_DOWNLOADS, ACCESS_CACHE_FILESYSTEM, ACCESS_CHECKIN_PROPERTIES, ACCESS_COARSE_LOCATION, ACCESS_COARSE_UPDATES, ACCESS_FINE_LOCATION, ACCESS_LOCATION_EXTRA_COMMANDS, ACCESS_MOCK_LOCATION, ACCESS_MTK_MMHW, ACCESS_NETWORK_STATE, ACCESS_PROVIDER, ACCESS_SERVICE, ACCESS_SHARED_DATA, ACCESS_SUPERUSER, ACCESS_SURFACE_FLINGER, ACCESS_WIFI_STATE, activityCalled, ACTIVITY_RECOGNITION, ACCOUNT_MANAGER, ADD_VOICEMAIL, ANT, ANT_ADMIN, AUTHENTICATE_ACCOUNTS, AUTORUN_MANAGER_LICENSE_MANAGER, AUTORUN_MANAGER_LICENSE_SERVICE(.autorun), BATTERY_STATS, BILLING, BIND_ACCESSIBILITY_SERVICE, BIND_APPWIDGET, BIND_CARRIER_MESSAGING_SERVICE, BIND_DEVICE_ADMIN, BIND_DREAM_SERVICE, BIND_GET_INSTALL_REFERRER_SERVICE, BIND_INPUT_METHOD, BIND_NFC_SERVICE, BIND_goodwareTIFICATION_LISTENER_SERVICE, BIND_PRINT_SERVICE, BIND_REMOTEVIEWS, BIND_TEXT_SERVICE, BIND_TV_INPUT, BIND_VOICE_INTERACTION, BIND_VPN_SERVICE, BIND_WALLPAPER, BLUETOOTH, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEGED, BODY_SENSORS, BRICK, BROADCAST_PACKAGE_REMOVED, BROADCAST_SMS, BROADCAST_STICKY, BROADCAST_WAP_PUSH, C2D_MESSAGE, CALL_PHONE, CALL_PRIVILEGED, CAMERA, CAPTURE_AUDIO_OUTPUT, CAPTURE_SECURE_VIDEO_OUTPUT, CAPTURE_VIDEO_OUTPUT, CHANGE_COMPONENT_ENABLED_STATE, CHANGE_CONFIGURATION, CHANGE_DISPLAY_MODE, CHANGE_NETWORK_STATE, CHANGE_WIFI_MULTICAST_STATE, CHANGE_WIFI_STATE, CHECK_LICENSE, CLEAR_APP_CACHE, CLEAR_APP_USER_DATA, CONTROL_LOCATION_UPDATES, DATABASE_INTERFACE_SERVICE, DELETE_CACHE_FILES, DELETE_PACKAGES, DEVICE_POWER, DIAGgoodwareSTIC, DISABLE_KEYGUARD, DOWNLOAD_SERVICE, DOWNLOAD_WITHOUT_goodwareTIFICATION, DUMP, EXPAND_STATUS_BAR, EXTENSION_PERMISSION, FACTORY_TEST, FLASHLIGHT, FORCE_BACK, FULLSCREEN.FULL, GET_ACCOUNTS, GET_PACKAGE_SIZE, GET_TASKS, GET_TOP_ACTIVITY_INFO, GLOBAL_SEARCH, GOOGLE_AUTH, GOOGLE_PHOTOS, HARDWARE_TEST, INJECT_EVENTS, INSTALL_LOCATION_PROVIDER, INSTALL_PACKAGES, INSTALL_SHORTCUT, INTERACT_ACROSS_USERS, INTERNAL_SYSTEM_WINDOW, INTERNET, JPUSH_MESSAGE, KILL_BACKGROUND_PROCESSES, LOCATION_HARDWARE, MANAGE_ACCOUNTS, MANAGE_APP_TOKENS, MANAGE_DOCUMENTS, MAPS_RECEIVE, MASTER_CLEAR, MEDIA_BUTTON, MEDIA_CONTENT_CONTROL, MESSAGE, MODIFY_AUDIO_SETTINGS, MODIFY_PHONE_STATE, MOUNT_FORMAT_FILESYSTEMS, MOUNT_UNMOUNT_FILESYSTEMS, NFC, PERSISTENT_ACTIVITY, PERMISSION, PERMISSION_RUN_TASKS, PLUGIN, PROCESS_OUTGOING_CALLS, READ, READ_ATTACHMENT, READ_AVESTTINGS, READ_CALENDAR, READ_CALL_LOG, READ_CONTACTS, READ_CONTENT_PROVIDER, READ_DATA, READ_DATABASES, READ_EXTERNAL_STORAGE, READ_FRAME_BUFFER, READ_GMAIL, READ_GSERVICES, READ_HISTORY_BOOKMARKS, READ_INPUT_STATE, READ_LOGS, READ_MESSAGES, READ_OWNER_DATA, READ_PHONE_STATE, READ_PROFILE, READ_SETTINGS, READ_SMS, READ_SOCIAL_STREAM, READ_SYNC_SETTINGS, READ_SYNC_STATS, READ_USER_DICTIONARY, READ_VOICEMAIL, REBOOT, RECEIVE, RECEIVE_BOOT_COMPLETED, RECEIVE_MMS, RECEIVE_SIGNED_DATA_RESULT, RECEIVE_SMS, RECEIVE_USER_PRESENT, RECEIVE_WAP_PUSH, RECORD_AUDIO, REORDER_TASKS, RESPOND, RESTART_PACKAGES, REQUEST, SDCARD_WRITE, SEND, SEND_RESPOND_VIA_MESSAGE, SEND_SMS, SET_ACTIVITY_WATCHER, SET_ALARM, SET_ALWAYS_FINISH, SET_ANIMATION_SCALE, SET_DEBUG_APP, SET_ORIENTATION, SET_POINTER_SPEED, SET_PREFERRED_APPLICATIONS, SET_PROCESS_LIMIT, SET_TIME, SET_TIME_ZONE, SET_WALLPAPER, SET_WALLPAPER_HINTS, SIGNAL_PERSISTENT_PROCESSES, STATUS_BAR, STORAGE, SUBSCRIBED_FEEDS_READ, SUBSCRIBED_FEEDS_WRITE, SYSTEM_ALERT_WINDOW, TRANSMIT_IR, UNINSTALL_SHORTCUT, UPDATE_DEVICE_STATS, USES_POLICY_FORCE_LOCK, USE_CREDENTIALS, USE_FINGERPRINT, USE_SIP, VIBRATE, WAKE_LOCK, WRITE, WRITE_APN_SETTINGS, WRITE_AVSETTING, WRITE_CALENDAR, WRITE_CALL_LOG, WRITE_CONTACTS, WRITE_DATA, WRITE_DATABASES, WRITE_EXTERNAL_STORAGE, WRITE_GSERVICES, WRITE_HISTORY_BOOKMARKS, WRITE_INTERNAL_STORAGE, WRITE_MEDIA_STORAGE, WRITE_OWNER_DATA, WRITE_PROFILE, WRITE_SECURE_SETTINGS, WRITE_SETTINGS, WRITE_SMS, WRITE_SOCIAL_STREAM, WRITE_SYNC_SETTINGS, WRITE_USER_DICTIONARY, WRITE_VOICEMAIL, Ljava/lang/reflect/Method;->invoke, Ljavax/crypto/Cipher;->doFinal, Ljava/lang/Runtime;->exec, Ljava/lang/System;->load, Ldalvik/system/DexClassLoader;->loadClass, Ljava/lang/System;->loadLibrary, Ljava/net/URL;-

>openConnection, Landroid/hardware/Camera;->open, Landroid/hardware/Camera;->takePicture, Landroid/telephony/SmsManager;->sendMultipartTextMessage, Landroid/telephony/SmsManager;->sendTextMessage, Landroid/media/AudioRecord;->startRecording, Landroid/telephony/TelephonyManager;->getCellLocation, Lcom/google/android/gms/location/LocationClient;->getLastLocation, Landroid/location/LocationManager;->getLastKgoodwarewnLocation, Landroid/telephony/TelephonyManager;->getDeviceId, Landroid/content/pm/PackageManager;->getInstalledApplications, Landroid/content/pm/PackageManager;->getInstalledPackages, Landroid/telephony/TelephonyManager;->getLine1Number, Landroid/telephony/TelephonyManager;->getNetworkOperator, Landroid/telephony/TelephonyManager;->getNetworkOperatorName, Landroid/telephony/TelephonyManager;->getNetworkCountryIso, Landroid/telephony/TelephonyManager;->getSimOperator, Landroid/telephony/TelephonyManager;->getSimOperatorName, Landroid/telephony/TelephonyManager;->getSimCountryIso, Landroid/telephony/TelephonyManager;->getSimSerialNumber, Lorg/apache/http/impl/client/DefaultHttpClient;->execute,

The model performance will not be optimal if the number of attributes is 241, including one label. Several factors cause this, namely, not all of the 241 features have strong relevance to the target label or feature. Second, the greater the number of irrelevant features in the dataset, the classification model will be optimal in terms of performance and computing time. Based on these problems, at this stage, feature reduction will be carried out using RFE with based models such as Gradient Boosting (GB), K-Nearest Neighbour (KNN), Decision Tree C4.5 (DT C4.5), Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP).

The result of feature reduction using RFE from several base models is a subset of new features whose number can vary between base models. The following are the results of feature reduction using RFE from each base model used in Table 3.

Table 3
Recursive feature elimination cross validation results for each base model

Base Model	Number of Original Features	Number of Selected Features
Gradient Boosting	241 (All Features showed in Table 2)	97 (All selected features showed here)
K-Nearest Neighbor	241 (All Features showed in Table 2)	177 (All selected features showed here)
Decision Tree C4.5	241 (All Features showed in Table 2)	47 (All selected features showed here)
Random Forest	241 (All Features showed in Table 2)	58 (All selected features showed here)
Support Vector Machine	241 (All Features showed in Table 2)	38 (All selected features showed here)
Multi-Layer Perceptron	241 (All Features showed in Table 2)	68 (All selected features showed here)

Table 3 shows that each base model used in RFECV feature reduction produces a different number of selected features. The first base model of Gradient Boosting produces 97 selected features; this means that 144 features in the dataset are eliminated. The second base model, K-Nearest Neighbour, produces 177 selected features, so the number of features reduced is 64. The third base model of Decision Tree C4.5 produces 47 selected features, which means that 194 features in the dataset are eliminated. The fourth base model of Random Forest produces 58 selected features, so the number of features reduced is 183 features. The fifth base model, the Support Vector Machine, produces 38 selected features; this means that 203 features in the dataset have been eliminated. Finally, the sixth base model of the Multi-Layer Perceptron produces 68 selected features, so the number of features reduced is 173 features.

After obtaining the selected features from the six base models used in the RFECV feature reduction, the next step is the feature filtering process using the intersection filter model. It works by taking several intersecting features from the selected features of the six base models. After that, the method added features ranking one from several base models. The Intersection Filter results can be seen in Table 4 below.

Table 4
Selected features results from intersection filtering based on RFECV

Selected Features Name
ACCESS_PROVIDER, ACCESS_SURFACE_FLINGER, BLUETOOTH_ADMIN, CAPTURE_AUDIO_OUTPUT, GET_TOP_ACTIVITY_INFO, JPUSH_MESSAGE, LOCATION_HARDWARE, MEDIA_CONTENT_CONTROL, MODIFY_PHONE_STATE, READ_CONTENT_PROVIDER, READ_MESSAGES, READ_PROFILE, READ_SYNC_STATS, RECEIVE, RECEIVE_MMS, RECEIVE_USER_PRESENT, REQUEST, SET_ACTIVITY_WATCHER, SET_WALLPAPER_HINTS, TRANSMIT_IR, WAKE_LOCK, WRITE_GSERVICES, WRITE_SMS, WRITE_SOCIAL_STREAM, Ljavax/crypto/Cipher;->doFinal, Ljava/lang/Runtime;->exec, Ljava/lang/System;->load, Ldalvik/system/DexClassLoader;->loadClass, Landroid/hardware/Camera;->open, Landroid/media/AudioRecord;->startRecording, Landroid/telephony/TelephonyManager;->getCellLocation, Lcom/google/android/gms/location/LocationClient;->getLastLocation, Landroid/telephony/TelephonyManager;->getDeviceId, Landroid/content/pm/PackageManager;->getInstalledPackages, Landroid/telephony/TelephonyManager;->getNetworkOperatorName, Landroid/telephony/TelephonyManager;->getSimCountryIso,

Table 4 shows that after carrying out the feature filtering process using the intersection filter model from the six base models, 36 selected features were obtained. The names of the resulting selected features can be seen in Table 4 above. The dataset resulting from the Intersection Filter process based on RFECV will be subjected to a modelling process using several classifiers. The single classifier models used include K-Nearest Neighbour (KNN), Decision Tree C4.5 (DT C4.5), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Meanwhile, the ensemble classifier models used are Gradient Boosting (GB) and Random Forest (RF). The following Figure 2 is a comparison of the number of features between Original Set, RFECV and Intersection based on RFECV.

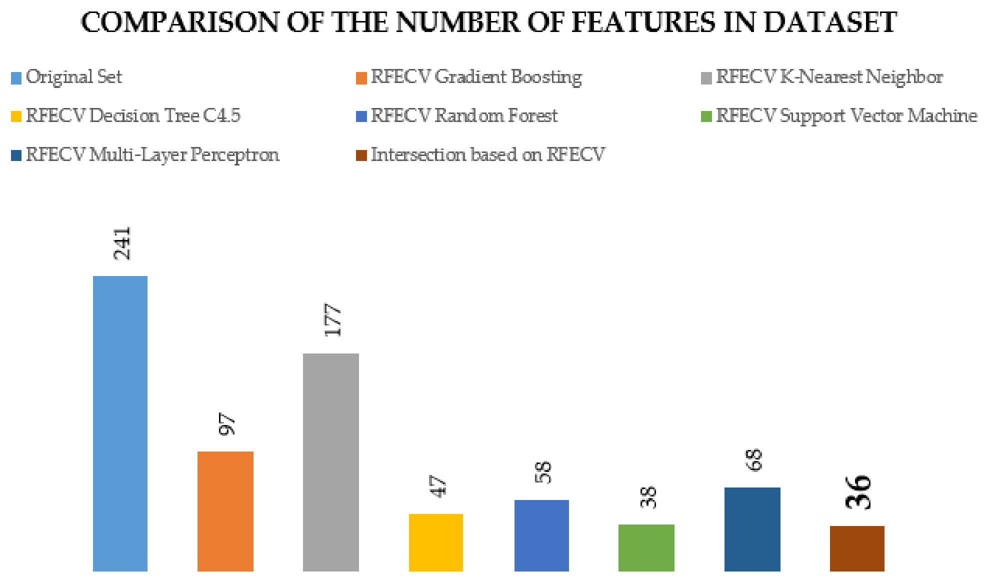

Fig. 2. Comparison of the number features in dataset

In general, from the dataset used, the combination of wrapping and filtering using Intersection Filtering based on RFECV (IF-RFECV) can produce fewer features than using the wrapping process alone. The feature results from IF-RFECV intersect with the features produced by various RFECV-based classification models. Based on Figure 2 above, the feature reduction model using Intersection based on RFECV can produce a smaller subset of features. This will certainly greatly influence the performance of the classification model that will be used. Apart from that, the subset of features produced are features that significantly influence the label or target of each base model.

3.2 Classification and Evaluation Model

After feature reduction, the next stage is evaluating and comparing classification models. In this research, two types of classification models are used, namely single classifier and ensemble classifier models. The single classifier models used include K-Nearest Neighbour (KNN), Decision Tree C4.5 (DT C4.5), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Meanwhile, the ensemble classifier models used are Gradient Boosting (GB) and Random Forest (RF). Meanwhile, the evaluation parameters that will be measured are Accuracy, Precision, Recall, F1 Score and Time Processing (seconds) . This section will also compare classification models for each indicator used.

Table 5 and Figure 3 are the results of the evaluation between classification models based on accuracy parameters.

Table 5
Accuracy values comparison of classification model

	DT C4.5	Gradient Boosting	KNN	SVM	Random Forest	MLP
Original Features	0.958	0.953	0.957	0.951	0.965	0.96
RFECV	0.967	0.963	0.958	0.958	0.975	0.931
Intersection based on RFECV (Proposed Method)	0.983	0.973	0.979	0.956	0.987	0.976

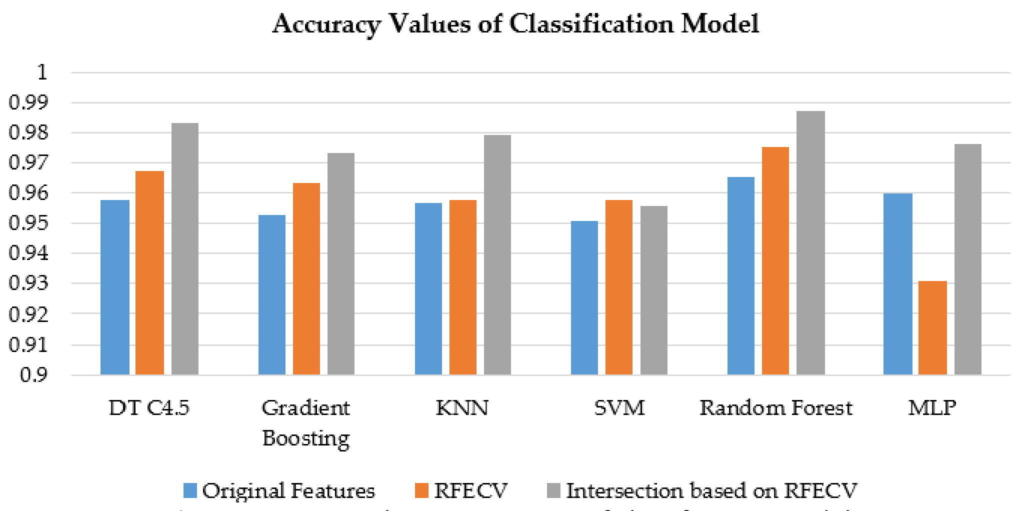

Fig. 3. Accuracy values comparison of classification model

Table 5 and Figure 3 show the evaluation model using the accuracy value parameters. The value of the intersection filtering technique based on recursive feature elimination cross-validation (IF-RFECV) can increase the accuracy of the classification model, including both single classifier and ensemble classifier models. The first model, DT C4.5, produced an accuracy value of 0.983 or 98.3%. In this first model, the resulting accuracy value is better than only the original features and RFECV. The second gradient-boosting model can produce an accuracy value of 0.973 or 97.3%. In this second model, the resulting ac-curacy values are better than just the original features and RFECV. The third

KNN model produced an accuracy value of 0.979 or 97.9%. In this third model, the resulting accuracy values are better than just the original features and RFECV. The fourth SVM model can produce an accuracy value of 0.956 or 95.6%. In this fourth model, the resulting accuracy value is almost equivalent to the original features and RFECV. The fifth Random Forest model produced an accuracy value of 0.987 or 98.7%. In this fifth model, the resulting ac-curacy values are better than only the original features and RFECV. The sixth MLP model can produce an accuracy value of 0.976 or 97.6%. In this sixth model, the resulting accuracy values are better than the original features and RFECV. In general, feature reduction using Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) can produce better accuracy values for the classification model compared to original features and RFECV alone.

Table 6 and Figure 4 are the results of the evaluation between classification models based on precision parameters.

Table 6
Precision values comparison of classification model

	DT C4.5	Gradient Boosting	KNN	SVM	Random Forest	MLP
Original Features	0.958	0.947	0.945	0.916	0.959	0.958
RFECV	0.968	0.95	0.949	0.912	0.968	0.956
Intersection based on RFECV (Proposed Method)	0.978	0.955	0.967	0.865	0.975	0.96

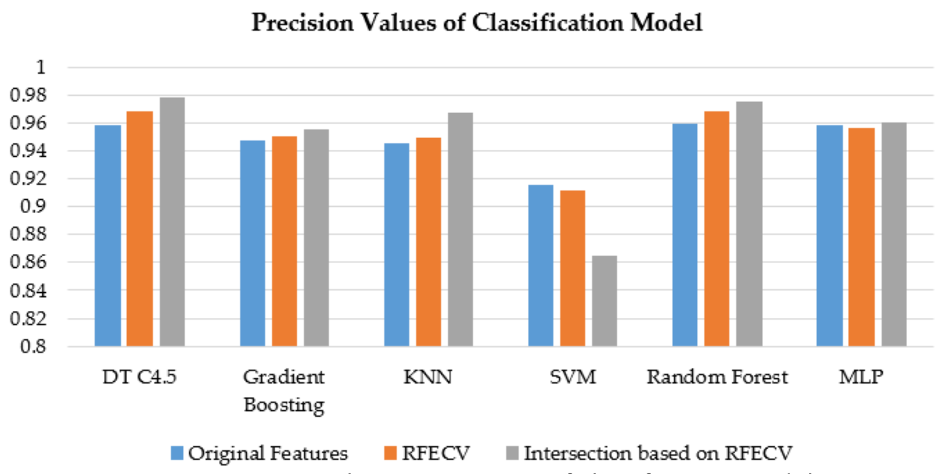

Fig. 4. Precision values comparison of classification model

Table 6 and Figure 4 show the precision value parameters evaluation model. It can be seen that the value of the Intersection Filtering technique based on Recursive Feature Elimination cross-validation (IF-RFECV) can increase the precision of the entire classification model used, both single classifier and ensemble classifier models. The first model, DT C4.5, produced a precision value of 0.978 or 97.8%. In this first model, the resulting precision value is better than only the original features and RFECV. The second Gradient Boosting model can produce a precision value of 0.955 or 95.5%. In this second model, the resulting precision value is better than only the original features and RFECV. The third KNN model produced a precision value of 0.967 or 96.7%. In this third model, the resulting precision value is better than only the original features and RFECV. The fourth SVM model can produce a precision value of 0.865 or 85.5%. In this fourth model, the resulting precision value is slightly lower than the original features and RFECV. The fifth Random Forest model can produce a precision value of 0.975 or 97.5%. In this fifth model, the resulting precision value is better than only the original features and RFECV. The sixth MLP model can produce a precision value of 0.96 or 96%.

This sixth model's resulting precision value is better than the original features and RFECV. In general, feature reduction using Intersection Filtering based on Recursive Feature Elimination Cross Validation (IF-RFECV) can produce precision values in the classification model that are better than the original features and also RFECV alone, except for the third model, namely SVM, which results below both.

Table 7 and Figure 5 are the results of the evaluation between classification models based on recall parameters.

Table 7
Recall values comparison of classification model

	DT C4.5	Gradient Boosting	KNN	SVM	Random Forest	MLP
Original Features	0.932	0.914	0.928	0.934	0.957	0.933
RFECV	0.923	0.925	0.929	0.942	0.96	0.942
Intersection based on RFECV (Proposed Method)	0.95	0.928	0.939	0.945	0.979	0.935

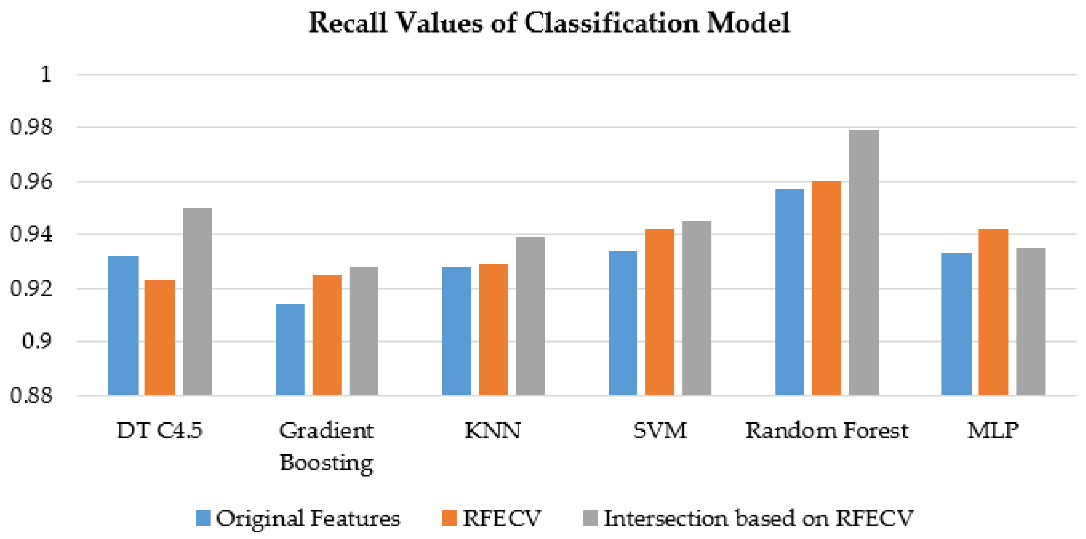

Fig. 5. Recall values comparison of classification model

Table 7 and Figure 5 show the evaluation model using the recall value parameters. It can be seen that the value of the Intersection Filtering technique based on Recursive Feature Elimination cross-validation (IF-RFECV) can increase the recall of the entire classification model used—both single classifier and ensemble classifier models. The first model, DT C4.5, produced a recall value of 0.95 or 95%. In this first model, the resulting recall value is better than only the original features and RFECV. The second gradient-boosting model can produce a recall value of 0.928 or 92.8%. In this second model, the resulting re-call value is better than just the original features and RFECV. The third KNN model produced a recall value 0.939, or 93.9%. In this third model, the resulting recall value is better than only the original features and RFECV. The fourth SVM model can produce a recall value of 0.945 or 94.5%. In this fourth model, the resulting recall value is better than only the original features and RFECV. The fifth Random Forest model produced a recall value 0.979 or 97.9%. In this fifth model, the resulting recall value is better than only the original features and RFECV. The sixth MLP model can produce a recall value of 0.935 or 93.5%. This sixth model's resulting recall value is slightly lower than the original features and RFECV. In general, feature reduction using Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) can produce recall values for the classification model that are better than the original features and RFECV alone, except for the sixth model, namely MLP, which produces results below both.

Table 8 and Figure 6 are the evaluation results between classification models based on the f1-score parameter.

Table 8
F-1 score comparison of classification model

	DT C4.5	Gradient Boosting	KNN	SVM	Random Forest	MLP
Original Features	0.947	0.925	0.936	0.924	0.956	0.945
RFECV	0.948	0.939	0.942	0.925	0.965	0.935
Intersection based on RFECV (Proposed Method)	0.971	0.945	0.958	0.909	0.984	0.952

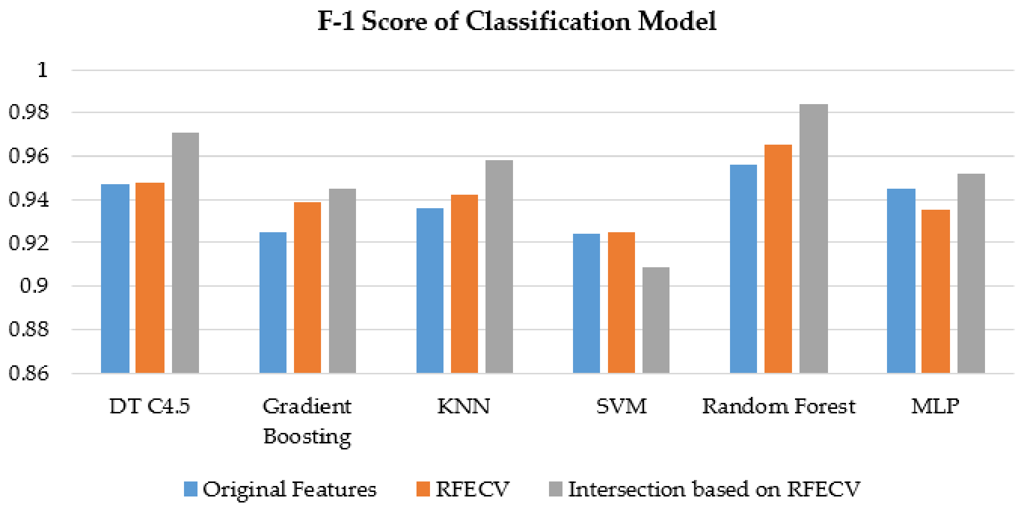

Fig. 6. F-1 score comparison of classification model

Table 8 and Figure 6 show the evaluation model using the f-1 score parameters. The value of the intersection filtering technique based on recursive feature elimination cross-validation (IF-RFECV) can increase the f-1 score of the entire classification model using both single classifier and ensemble classifier models. The first model, DT C4.5, produced an f-1 score of 0.971 or 97.1%. The resulting f-1 score is better than only the original features and RFECV in this first model. The second gradient-boosting model can produce an f-1 score of 0.945 or 94.5%. In this second model, the resulting f-1 score is better than just the original features and RFECV. The third KNN model produced an f-1 score of 0.958 or 95.8%. In this third model, the resulting f-1 score is better than just the original features and RFECV. The fourth SVM model can produce an f-1 score of 0.909 or 90.9%. In this fourth model, the resulting f-1 score is slightly lower than the original features and RFECV. The fifth Random Forest model produced an f-1 score of 0.984 or 98.4%. In this fifth model, the resulting f-1 score is better than only the original features and RFECV. The sixth MLP model can produce an f-1 score of 0.952 or 95.2%. In this sixth model, the resulting f-1 score is better than the original features and RFECV. In general, feature reduction using Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) can produce an f-1 score for the classification model that is better than the original features and RFECV alone, except for the third model, namely SVM, which pro-duces the same results, below both.

Table 9 and Figure 7 are the results of the evaluation between classification models based on the Processing Time parameter.

Table 9
Processing time (in seconds) comparison of classification model

	DT C4.5	Gradient Boosting	KNN	SVM	Random Forest	MLP
Original Features	1.9	25.8	8.6	9.8	8.1	14.1
RFECV	906	1277.8	547.9	547.4	552.6	526.5
Intersection based on RFECV (Proposed Method)	907.2	1309.7	551.6	560.9	564.1	557.8

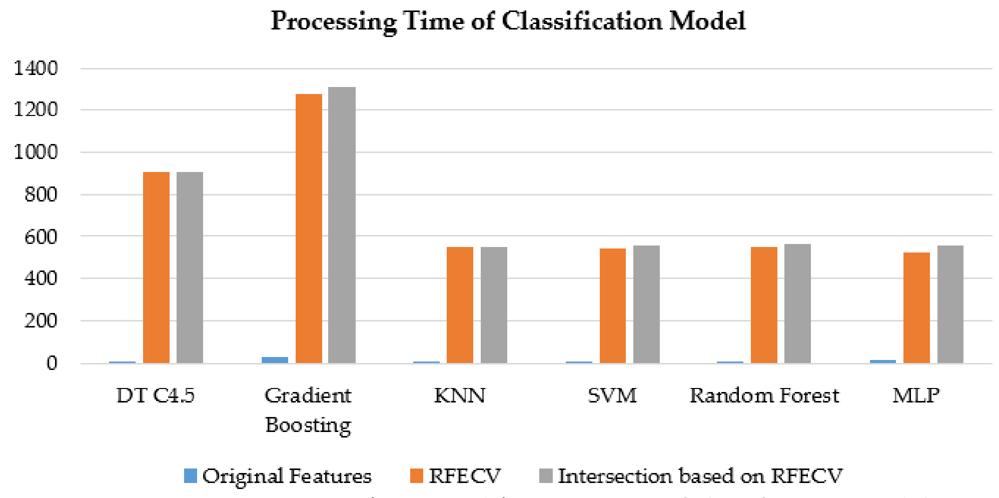

Fig. 7. Processing time (in seconds) comparison of classification model

Based on the evaluation using the Processing Time parameters in Table 9 and Figure 7 above, it can be seen that the value of the Intersection Filtering technique based on Recursive Feature Elimination Cross Validation (IF-RFECV) produces a processing time value which is of course slightly longer than RFECEV. This is due to how IF-RFECV works, where it has to wait for the RFECV results before selecting the feature slices. Of course, this will increase the processing time. Combining wrapping and filtering using Intersection Filtering based on RFECV (IF-RFECV) can produce fewer features than just using the wrapping process alone or the original features set. The reduced feature results from IF-RFECV intersect with the features produced by various RFECV-based classification models. The reduction feature using Intersection Filtering based on RFECV (IF-RFECV) can produce better performance for most classification models compared to RFECV alone or the original features set.

4. Conclusions

Based on the research that has been carried out, malware detection on the Android system is an essential aspect of maintaining data security and user privacy. One technology that can be used is machine learning. In this research, a machine learning model is proposed to detect malware on Android. The proposed Intersection Filtering method based on RFECV (IF-RFECV) can produce fewer features and correlate with the label or target class. Overall, feature reduction using Intersection Filtering based on Recursive Feature Elimination Cross Validation (IF-RFECV) can produce accuracy, precision, recall and f-1 scores on the classification model that are better than original features or RFECV alone. The processing time carried out by Intersection Filtering based on Recursive Feature Elimination cross-validation (IF-RFECV) is similar to original features or RFECV alone. With the increase in results, this model can be used well in detecting malware on the Android operating system. This research has limitations: it can only detect two classes, namely Malware and Good-ware.

Therefore, in future research, a detection model with multi-classes and types of malwares can be developed in detail using a machine learning model.

Acknowledgment
The researcher would like to sincerely thank the Department of Research and Community Service, and Faculty of Computer Science Universitas Amikom Yogyakarta, for supporting and funding this research.

References
[1] Chemmakha, Mohammed, Omar Habibi, and Mohamed Lazaar. "Improving machine learning models for malware detection using embedded feature selection method." *IFAC-PapersOnLine* 55, no. 12 (2022): 771-776. https://doi.org/10.1016/j.ifacol.2022.07.406

[2] Karabatak, Murat, and Twana Mustafa. "Performance comparison of classifiers on reduced phishing website dataset." In *2018 6th international symposium on digital forensic and security (ISDFS)*, pp. 1-5. IEEE, 2018. https://doi.org/10.1109/ISDFS.2018.8355357

[3] Liu, Kaijun, Shengwei Xu, Guoai Xu, Miao Zhang, Dawei Sun, and Haifeng Liu. "A review of android malware detection approaches based on machine learning." *IEEE access* 8 (2020): 124579-124607. https://doi.org/10.1109/ACCESS.2020.3006143

[4] Mahindru, Arvind, and Amrit Lal Sangal. "MLDroid—framework for Android malware detection using machine learning techniques." *Neural Computing and Applications* 33, no. 10 (2021): 5183-5240. https://doi.org/10.1007/s00521-020-05309-4

[5] Li, Chaoran, Xiao Chen, Derui Wang, Sheng Wen, Muhammad Ejaz Ahmed, Seyit Camtepe, and Yang Xiang. "Backdoor attack on machine learning based android malware detectors." *IEEE Transactions on dependable and secure computing* 19, no. 5 (2021): 3357-3370. https://doi.org/10.1109/TDSC.2021.3094824

[6] Lee, Jaehyeong, Hyuk Jang, Sungmin Ha, and Yourim Yoon. "Android malware detection using machine learning with feature selection based on the genetic algorithm." *Mathematics* 9, no. 21 (2021): 2813. https://doi.org/10.3390/math9212813

[7] Senanayake, Janaka, Harsha Kalutarage, and Mhd Omar Al-Kadri. "Android mobile malware detection using machine learning: A systematic review." *Electronics* 10, no. 13 (2021): 1606. https://doi.org/10.3390/electronics10131606

[8] Abdullah, Talal AA, Waleed Ali, and Rawad Abdulghafor. "Empirical study on intelligent android malware detection based on supervised machine learning." *International Journal of Advanced Computer Science and Applications* 11, no. 4 (2020). https://doi.org/10.14569/IJACSA.2020.0110429

[9] Wu, Qing, Xueling Zhu, and Bo Liu. "A survey of android malware static detection technology based on machine learning." *Mobile Information Systems* 2021, no. 1 (2021): 8896013. https://doi.org/10.1155/2021/8896013

[10] Bhat, Parnika, Sunny Behal, and Kamlesh Dutta. "A system call-based android malware detection approach with homogeneous & heterogeneous ensemble machine learning." *Computers & Security* 130 (2023): 103277. https://doi.org/10.1016/j.cose.2023.103277

[11] Molina-Coronado, Borja, Usue Mori, Alexander Mendiburu, and Jose Miguel-Alonso. "Towards a fair comparison and realistic evaluation framework of android malware detectors based on static analysis and machine learning." *Computers & Security* 124 (2023): 102996. https://doi.org/10.1016/j.cose.2022.102996

[12] Gera, Tanya, Jaiteg Singh, Abolfazl Mehbodniya, Julian L. Webber, Mohammad Shabaz, and Deepak Thakur. "Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware." *Security and Communication Networks* 2021, no. 1 (2021): 7035233. https://doi.org/10.1155/2021/7035233

[13] Tahtaci, Burak, and Beyzanur CANBAY. "Android malware detection using machine learning." In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ASYU50717.2020.9259834

[14] El Fiky, Ahmed Hashem, Ayman Elshenawy, and Mohamed Ashraf Madkour. "Detection of android malware using machine learning." In *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 9-16. IEEE, 2021. https://doi.org/10.1109/MIUCC52538.2021.9447661

[15] Islam, Rejwana, Moinul Islam Sayed, Sajal Saha, Mohammad Jamal Hossain, and Md Abdul Masud. "Android malware classification using optimum feature selection and ensemble machine learning." *Internet of Things and Cyber-Physical Systems* 3 (2023): 100-111. https://doi.org/10.1016/j.iotcps.2023.03.001

[16] Mustaqim, Adi Zaenul, Sumarni Adi, Yoga Pristyanto, and Yuli Astuti. "The effect of recursive feature elimination with cross-validation (RFECV) feature selection algorithm toward classifier performance on credit card fraud detection." In *2021 International conference on artificial intelligence and computer science technology (ICAICST)*, pp. 270-275. IEEE, 2021. https://doi.org/10.1109/ICAICST53116.2021.9497842
[17] Borah, P., and D. K. Bhattacharyya. "TUANDROMD (Tezpur university android malware dataset)." *UCI Machine Learning Repository* (2023).
[18] Borah, Parthajit, D. K. Bhattacharyya, and J. K. Kalita. "Malware dataset generation and evaluation." In *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/CICT51604.2020.9312053
[19] Sanjay, Amrita, H. Vinayak Nair, Sruthy Murali, and K. S. Krishnaveni. "A data mining model to predict breast cancer using improved feature selection method on real time data." In *2018 international conference on advances in computing, communications and informatics (ICACCI)*, pp. 2437-2440. IEEE, 2018. https://doi.org/10.1109/ICACCI.2018.8554450
[20] Riajuliislam, Md, Khandakar Zahidur Rahim, and Antara Mahmud. "Prediction of thyroid disease (hypothyroid) in early stage using feature selection and classification techniques." In *2021 International conference on information and communication technology for sustainable development (ICICT4SD)*, pp. 60-64. IEEE, 2021. https://doi.org/10.1109/ICICT4SD50815.2021.9397052
[21] Adorada, Amazona, Panji Wisnu Wirawan, and Kabul Kurniawan. "The comparison of feature selection methods in software defect prediction." In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ICICoS51170.2020.9299022
[22] Aziz, Mohd Zafran Abdul, and Koji Okamura. "A security trending review on software define network (SDN)." *Journal of Advanced Research in Computing and Applications* 6, no. 1 (2017): 9-19.
[23] Morshidi, Azizan, Noor Syakirah Zakaria, Mohammad Ikhram Mohammad Ridzuan, Rizal Zamani Idris, Azueryn Annatassia Dania Aqeela, and Mohamad Shaukhi Mohd Radzi. "Artificial Intelligence and Islam: A Bibiliometric-Thematic Analysis and Future Research Direction." *Semarak International Journal of Machine Learning* 1, no. 1 (2024): 41-58. https://doi.org/10.37934/sijml.1.1.4158
[24] Ong, Siew Har, Sai Xin Ni, and Ho Li Vern. "Dimensions Affecting Consumer Acceptance towards Artificial Intelligence (AI) Service in the Food and Beverage Industry in Klang Valley." *Semarak International Journal of Machine Learning* 1, no. 1 (2024): 20-30. https://doi.org/10.37934/sijml.1.1.2030
[25] Amutha, S., S. Santhi, And L. Monicca. "Decision support system for diagnosing disease." *International Journal of Pharmaceutical Research (09752366)* 11, no. 1 (2019). https://doi.org/10.31838/ijpr/2019.11.01.090