INTEGRATED INFORMATION GAIN WITH EXTRA TREE ALGORITHM FOR FEATURE PERMISSION ANALYSIS IN ANDROID MALWARE CLASSIFICATION

HOWIDA ABUBAKER AL-KAAF

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JUNE 2022

DEDICATION

I dedicated this thesis to my family. A special feeling of gratitude to my beloved mother, siblings, and friends for their endless support and engorgement throughout my academic journey.

I also dedicated this thesis to the memory of my father, who always believed in my ability to be successful in the academic area. You are gone but your belief in me has made this journey possible.

ACKNOWLEDGEMENT

First and foremost, all praise to Allah -SWT- for his strengths and blessings and peace, and blessings be upon his Messenger, Mohammed (Peace Be Upon Him).

I would like to express my deepest gratitude to my main supervisor, **Dr. Aida bint Ali**, for her encouragement, guidance, critics and friendship, enormous patience, and generous support throughout this study. She taught me uncountable lessons on academic research as well as life and patiently guided me towards my goals. Without her continued support and interest, this thesis would not have been the same as presented here. Also thanks to my co-supervisor, **Dr. Shafaatunnur Hassan** for her guidance and support.

Moreover, I would like to acknowledge this thesis to the memory of my previous deceased supervisor Almarhumah **Prof. Dr. Siti Mariyam Shamsuddin** who encouraged me during my study.

My sincere appreciation also extends to all my friends, colleagues, and staff in UTM who have provided assistance at various occasions.

Last but not least, my deepest gratitude to my beloved family. My special thanks go to my beloved mother, for her endless love, patience, and prayers. And thanks to my siblings for their support and prayers. And thanks to my sister for accompanying me in my journey in UTM.

ABSTRACT

The rapid growth of free applications in the android market has led to the fast spread of malware apps since users store their sensitive personal information on their mobile devices when using those apps. The permission mechanism is designed as a security layer to protect the android operating system by restricting access to local resources of the system at installation time and run time for updated versions of the android operating system. Even though permissions provide a secure layer to users, they can be exploited by attackers to threaten user privacy. Consequently, exploring the patterns of those permissions becomes necessary to find the relevant permission features that contribute to classifying android apps. However, with the era of big data and the rapid explosion of malware along with many unnecessary requested permissions, it has become a challenge to recognize the patterns of permissions from these data due to the irrelevant and redundant features that affect the classification performance and increase the complexity cost overhead. Ensemble-based Extra Tree - Feature Selection (FS-EX) algorithm was proposed in this study to explore the permission patterns by selecting a minimal-sized subset of highly discriminant permission features capable of discriminating against malware samples from non-malware samples. The integrated Information Gain with Ensemble-based Extra Tree - Feature Selection (FS-IGEX) algorithm is proposed to assign weight values to permission features instead of binary values to determine the impact of weighted attribute variables on the classification performance. The two proposed methods based on Ensemble Extra Tree Feature Selection were evaluated on five datasets with various sample sizes and feature space using nine machine learning classifiers. Comparison studies were carried out between FS-EX subsets and the dataset of Full Permission features (FP) and the two approaches of the FS-IGEX method - the Permission-Binary (PB) approach and the Permission-Weighted (PW) approach. The permissions with PB were represented with binary values, whereas permissions with PW were represented with weighted values. The results demonstrated that the approach with the FS-EX was promising in obtaining the most prominent permission features related to the class target and attaining the same or close classification results in terms of accuracy with the highest accuracy mean of 96%, as compared to the FP. In addition, the PW approach of the FS-IGEX method had highly influential weighted permission features that could classify apps as malware and non-malware with the highest accuracy mean of 93%, compared to the PB approach of the FS-IGEX method and the FP.

ABSTRAK

Pertumbuhan pesat aplikasi percuma dalam pasaran android telah membawa kepada penyebaran pantas aplikasi perisian hasad yang cepat kerana pengguna menyimpan maklumat peribadi mereka pada peranti mudah alih mereka apabila menggunakan aplikasi tersebut. Mekanisme kebenaran direka bentuk sebagai lapisan keselamatan untuk melindungi system pengendalian android dengan mengehadkan akses kepada sumber sistem tempatan pada pemasangan masa dan masa jalankan untuk versi terkini sistem pengendalian android. Walaupun kebenaran menyediakan lapisan selamat kepada pengguna, ia boleh dieksploitasi oleh penyerang untuk mengancam privasi pengguna. Oleh itu, penerokaan corak kebenaran tersebut menjadi satu keperluan untuk mencari ciri-ciri kebenaran yang berkaitan yang menyumbang kepada pengklasifikasian android. Walau bagaimanapun, dengan era data besar dan ledakan pesat perisian hasad bersama-sama dengan banyak permintaan yang tidak perlu kepada kebenaran, ia telah menjadi cabaran untuk mengenali corak data disebabkan oleh ciriciri yang tidak relevan dan berlebihan yang boleh mempengaruhi prestasi pengelasan dan meningkatkan kerumitan kos perbelanjaan. Algoritma Pokok Tambahan Berasaskan Ensemble - Pemilihan Ciri (FS-EX) telah dicadangkan dalam kajian ini untuk meneroka corak kebenaran dengan memilih subset bersaiz minimum ciri kebenaran yang sangat diskriminasi yang mampu mendiskriminasi sampel perisian hasad daripada sampel bukan perisian hasad. Algoritma bersepadu Pemilihan Maklumat dengan Pokok Tambahan Berasaskan Ensemble – Pemilihan Ciri (FS-IGEX) telah dicadangkan untuk memberikan nilai pemberat kepada ciri kebenaran dan bukannya nilai perduaan untuk menentukan kesan pembolehubah atribut berwajaran pada prestasi pengelasan. Kedua-dua kaedah yang dicadangkan berdasarkan Pemilihan Ciri Pokok Tambahan Ensemble dinilai pada lima set data dengan pelbagai saiz sampel dan ruang ciri menggunakan sembilan pengelas pembelajaran mesin. Kajian perbandingan telah dijalankan antara subset FS-EX dan set data ciri-ciri Kebenaran Penuh (FP) dan dua pendekatan kaedah FS-IGEX, pendekatan Binari Kebenaran (PB) dan pendekatan Wajaran Kebenaran (PW). Keizinan dengan PB diwakili dengan nilai binari, manakala kebenaran dengan PW diwakili dengan nilai wajaran. Keputusan menunjukkan bahawa pendekatan dengan FS-EX menjanjikan dalam mendapatkan ciri kebenaran yang paling menonjol berkaitan dengan sasaran kelas dan mencapai keputusan pengelasan yang sama atau hampir dari segi ketepatan dengan min ketepatan tertinggi sebanyak 96%, berbanding dengan FP. Selain itu, pendekatan PW bagi kaedah FS-IGEX mempunyai ciri wajaran kebenaran yang sangat berpengaruh yang boleh mengklasifikasikan aplikasi sebagai perisian hasad dan bukan perisian hasad dengan min ketepatan tertinggi sebanyak 93%, berbanding pendekatan PB bagi kaedah FS-IGEX dan FP.

TABLE OF CONTENTS

TITLE		PAGE
DECLARATION		**iii**
DEDICATION		**iv**
ACKNOWLEDGEMENT		**v**
ABSTRACT		**vi**
ABSTRAK		**vii**
TABLE OF CONTENTS		**viii**
LIST OF TABLES		**xii**
LIST OF FIGURES		**xv**
LIST OF ABBREVIATIONS		**xviii**
LIST OF APPENDICES		**xx**
CHAPTER 1	**INTRODUCTION**	**1**
1.1	Overview	1
1.2	Problem Background	3
1.3	Problem Statement	12
1.4	Aim of the Research	14
1.5	Objectives of the Research	14
1.6	Research Scope	15
1.7	Contributions of the Research	15
1.8	Significance of the Research	18
1.9	Organization of the Thesis	19
CHAPTER 2	**LITERATURE REVIEW**	**21**
2.1	Introduction	21
2.2	Android Architecture	22
	2.2.1 Android Applications	23
	2.2.2 Android Permissions	24
2.3	Issues and Challenges with Android Mobile Applications Classification	26

2.3.1 Mobile Devices Threats Related to Permissions 27
2.3.2 Feature Representation 31
2.4 Machine Learning Approaches for Malware Classification in Android Platform 31
2.5 Discussion 44
2.5.1 Permissions Features in Android Applications Classification 45
2.5.2 Feature Selection 48
2.5.3 Combined Feature Selection Methods 54
2.5.4 Weighting Features 54
2.6 Current Trends and Directions in Android Malware Classification 56
2.7 Summary 58

CHAPTER 3 INTRODUCTION 59
3.1 Introduction 59
3.2 Research Framework 61
3.3 The Design of the Proposed Ensemble-Based Extra Tree Feature Selection Method (FS-EX**)** 64
3.3.1 Extremely Randomized Tree Classifier (Extra Tree) 66
3.3.2 Feature Importance 68
3.4 The Design of the Proposed Method Integrated Information Gain with Ensemble-Based Extra Tree Feature Selection (FS-IGEX) 69
3.4.1 The Implementation of Information Gain (IG) Feature Selection Method 73
3.4.2 The Implementation of Ensemble-Based Extra Tree Feature Selection Method 73
3.4.3 Feature Representation Approach 74
3.5 Classifiers 75
3.6 Performance Evaluation 76
3.7 Statistical Significance Test 77
3.8 Datasets 79
3.9 Software and Hardware Requirement 83

3.10 Chapter Summary 84

CHAPTER 4 THE EFFECT OF PERMISSION FEATURES ON CLASSIFYING ANDROID APPS 85

4.1 Introduction 85

4.2 Experimental Setting 85

4.2.1 Experimental Procedures and Datasets Pre- processing 86

4.2.2 Classification Algorithms and Evaluation Metrics 86

4.3 Classification Results and Discussion 90

4.4 Findings 94

4.5 Summary 96

CHAPTER 5 THE PROPOSED ENSEMBLE - BASED EXTRA TREE FEATURE SELECTION ALGORITHM (FS-EX) 97

5.1 Introduction 97

5.2 Experimental Setting 98

5.2.1 Experimental Procedures and Datasets Pre-processing 98

5.2.2 The Implementation of Ensemble-Based Extra Tree Feature Selection 99

5.3 Feature Subsets 102

5.4 Classifier Evaluations 116

5.5 Classification Results and Discussion 116

5.6 Findings 133

5.7 Summary 137

CHAPTER 6 INTEGRATED INFORMATION GAIN WITH EXTRA TREE ALGORITHM (FS-IGEX) 139

6.1 Introduction 139

6.2 Experimental Setting 140

6.2.1 Experimental Procedures and Datasets Pre-processing 140

6.2.2 The Implementation of Information Gain Feature Selection (IG) 142

6.2.3 The Implementation of Ensemble-Based Extra Tree Feature Selection 149

6.2.4 Feature Subsets 152
6.3 Classification Results and Discussion 162
6.4 Findings 178
6.5 Summary 181

CHAPTER 7 CONCLUSION AND FUTURE WORK 183
7.1 Introduction 183
7.2 Research Summary 184
7.3 Research Findings 185
7.4 Future Work 189
7.5 Closing Remarks 190

REFERENCES 193

APPENDICES 213-223

LIST OF PUBLICATIONS 225

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 2.1	Some of the previous works on mobile applications classification	42
Table 3.1	Confusion matrix	77
Table 3.2	Some samples of MO dataset	81
Table 3.3	Some samples of Hybrid dataset with static and dynamic permission features	82
Table 3.4	Statistic on five different data sets used in experiments of the study	83
Table 4.1	Performance results of the classifiers on all five datasets	91
Table 4.2	Comparison with some existing related works	94
Table 5.1	Some permissions with their explanation and threats related to them	105
Table 5.2	The performance of (FS-EX) and (FP) of different tested machine-learning algorithms for MO dataset	119
Table 5.3	The result of Duncan's Multiple Range Statistical Test (DRMT) for (FS-EX) method on MO dataset based on accuracy mean	120
Table 5.4	The performance of (FS-EX) and (FP) of different tested machine-learning algorithms for Machine Learning 1 dataset	122
Table 5.5	The result of Duncan's Multiple Range Statistical Test (DRMT) for (FS-EX) method on Machine Learning 1 dataset based on accuracy mean	123
Table 5.6	The performance of (FS-EX) and (FP) of different tested machine-learning algorithms for Machine Learning 2 dataset	125

Table 5.7	The result of Duncan's Multiple Range Statistical Test (DRMT) for (FS-EX) method on Machine Learning 2 dataset based on accuracy mean	126
Table 5.8	The performance of (FS-EX) and (FP) of different tested machine-learning algorithms for Kaggle dataset	127
Table 5.9	The result of Duncan's Multiple Range Statistical Test (DRMT) for (FS-EX) method on Kaggle dataset based on accuracy mean	128
Table 5.10	The performance of (FS-EX) and (FP) of different tested machine-learning algorithms for Hybrid dataset	130
Table 5.11	The result of Duncan's Multiple Range Statistical Test (DRMT) for (FS-EX) method on Hybrid dataset based on accuracy mean	131
Table 6.1	The performance of FS-IGEX (PB &PW) approaches & full permissions (FP) on different tested machine learning algorithms for MO dataset	164
Table 6.2	The result of Duncan's Multiple Range Test (DRMT) for (FS-IGEX) method on MO dataset based on accuracy mean	165
Table 6.3	The performance of FS-IGEX (PB &PW) approaches & full permissions (FP) on different tested machine learning algorithms for Machine Learning1 dataset	167
Table 6.4	The result of Duncan's Multiple Range Test (DRMT) for (FS-IGEX) method on Machine Learning1 dataset based on accuracy mean	168
Table 6.5	The performance of FS-IGEX (PB &PW) approaches & full permissions (FP) on different tested machine learning algorithms for Machine Learning 2 dataset	170
Table 6.6	The result of Duncan's Multiple Range Test (DRMT) for (FS-IGEX) method on Machine Learning 2 dataset based on accuracy mean	171

Table 6.7	The performance of FS-IGEX (PB &PW) approaches & full permissions (FP) on different tested machine learning algorithms for Kaggle dataset	172
Table 6.8	The result of Duncan's Multiple Range Test (DRMT) for (FS-IGEX) method on Kaggle dataset based on accuracy mean	173
Table 6.9	The performance of FS-IGEX (PB &PW) approaches & full permissions (FP) on different tested machine learning algorithms for Hybrid dataset	175
Table 6.10	The result of Duncan's Multiple Range Test (DRMT) for (FS-IGEX) method on Hybrid dataset based on accuracy mean	176

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 1.1	The svpeng malware with a phishing window asking credit card details	4
Figure 2.1	Android architecture	23
Figure 2.2	Android app permissions types	26
Figure 2.3	Elements in literature study	46
Figure 3.1	Research Framework	60
Figure 3.2	The design of the proposed ensemble-based extra tree feature selection method (FS-EX)	65
Figure 3.3	The design of the proposed method of integrated information gain (IG) with ensemble-based extra tree feature selection method (FS-IGEX)	72
Figure 3.4	The process of collecting benign and malware apps and extracting their apk files & permissions keywords	80
Figure 4.1	The process of classifying android apps based on the permission features	87
Figure 5.1	The design framework of the proposed method – the ensemble-based extra tree feature selection (FS-EX)	101
Figure 5.2	The top 5, 10, and 20 important permission features for MO dataset obtained by using (FS-EX)	104
Figure 5.3	The top 5, 10, and 20 important permission features for Machine Learning 1 dataset obtained by using (FS-EX)	107
Figure 5.4	The top 5, 10, and 20 important permission features for Machine Learning 2 dataset obtained by using (FS-EX)	109
Figure 5.5	The top 5, 10, and 20 important permission features for kaggle dataset obtained by using (FS-EX)	111

Figure 5.6	The top 5, 10, and 20 important permission features for Hybrid dataset obtained by using (FS-EX)	114
Figure 6.1	The design framework of the proposed method – The improved weighted ensemble-based extra tree integrated with Information Gain feature selection (FS-IGEX)	141
Figure 6.2	The design framework of ranked features by using Information Gain (IG) feature selection	143
Figure 6.3	Top 5, 10, and 20 permission features ranked by using Information Gain (IG) for MO dataset	144
Figure 6.4	Top 5, 10, and 20 permission features ranked by using Information Gain (IG) for Machine Learning 1 dataset	145
Figure 6.5	Top 5, 10, and 20 permission features ranked by using Information Gain (IG) for Machine Learning 2 dataset	146
Figure 6.6	Top 5, 10, and 20 permission features ranked by using Information Gain (IG) for Kaggle dataset	147
Figure 6.7	Top 5, 10, and 20 permission features ranked by using Information Gain (IG) for Hybrid dataset	148
Figure 6.8	The process of computing weights of features	150
Figure 6.9	The process of classification task with the proposed method (FS-IGEX)	151
Figure 6.10	The top 5, 10, and 20 important permission features for MO dataset obtained by using (FS-IGEX)	153
Figure 6.11	The top 5, 10, and 20 important permission features for Machine Learning 1 dataset obtained by using (FS-IGEX)	155
Figure 6.12	The top 5, 10, and 20 important permission features for Machine Learning 2 dataset obtained by using (FS-IGEX)	156
Figure 6.13	The top 5, 10, and 20 important permission features for Kaggle dataset obtained by using (FS-IGEX)	159

Figure 6.14 The top 5, 10, and 20 important permission features for Hybrid dataset obtained by using (FS-IGEX) 162

LIST OF ABBREVIATIONS

ACC	-	Accuracy
AOFU	-	Ask-on-first-use
AOI	-	Ask-on-install
API	-	Application Programming Interface
APk	-	Android Application Package
APP	-	Application
CART	-	A Classification and Regression Tree
CSV	-	A Comma-Separated Values File
DT	-	Decision Tree Classifier
EX	-	Extra Tree Classifier
FNR	-	False Negative Rate
FP	-	Full Permission Features
FPR	-	False Positive Rate
FS-EX	-	Ensemble - Based Extra Tree - Feature Selection Algorithm
FS-IGEX	-	Integrated Information Gain with Extra Tree Algorithm
GPS	-	Global Positioning System
IG	-	Information Gain
K-NN	-	K-Nearest Neighbors Algorithm
IOT	-	Internet of Things
LDA	-	Linear Discriminant Analysis Classifier
LR	-	Logistic Regression Classifier
MMS	-	Multimedia Messaging Service
ML	-	Machine Learning
MLP	-	Multilayer Perceptron Classifier
NB	-	Naïve Bayes Classifier
NFC	-	Near Field Communication
PB	-	Permission-Binary
PCA	-	Principal Component Analysis
PDAs	-	Personal Digital Assistant
PW	-	Permission-Weighted

RF	-	Random Forest Classifier
RF	-	Rotation Forest
SMO	-	Sequential Minimal Optimization
SMS	-	Short Message Service
SVM	-	Support Vector Machine Classifier
TPR	-	True Positive Rate

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
Appendix A	Top Significant Permission Features Ranking using (FS-EX) for All Datasets	213
Appendix B	Top Significant Permission Features Ranking with their Final Weights using (FS-IGEX) for All Datasets	218

CHAPTER 1

INTRODUCTION

1.1 Overview

Mobile devices such as tablets, smartphones, and PDAs have become the most important technology in our daily life and the concern of many research communities and commercial industry as well. That consideration is due to the fast development in the hardware of mobile technology such as better processing power, larger wireless network bandwidth, and enhances capabilities of mobile devices. Additionally, the rapid growth of mobile platforms has led to the development of mobile applications as well such as social networks, mobile banking, mobile health, online learning applications, etc. (Idrees et al., 2017; Jaenke et al., 2021; López-Moranchel et al., 2021). Generally, mobile applications are 1.5 times faster than mobile websites and they perform actions much faster too. The data of applications are stored on mobile devices which could be restored in the blink of an eye. Examples of spreading apps as happened nowadays with mobile health apps for COVID-19 management such as contact tracing, managing patient care, and access to information about COVID-19 apps. The contact tracing app used Bluetooth technology to identify individuals who have been in close contact with patients who have been diagnosed with COVID-19 such as Covidsafe in Australia and Trace together in Singapore (Salehinejad et al., 2021). However, these technologies come up with many advantages, there are some new risks that must be considered. Sharing information with others through smartphone apps that store sensitive information could cause damage to the privacy of

the user (Appice et al., 2020). Moreover, mobile devices got the attention of attackers and hackers who take advantage of some features of communication-related to smartphones like Short Message Service (SMS), Multimedia Messaging Service (MMS), WIFI networks, and Global System for Mobile Communications (GSM) to inject the system with malicious software. Additionally, they exploit the vulnerabilities of software of the web browser and operating system and leak knowledge of users to launch an attack (Alazab, 2020). Malware is defined as a malicious application that intends to gain access to computer systems and network resources, disrupts computer operations, gathers personal information without the knowledge of the system's user, and puts the user's privacy at risk (Yan & Yan, 2018). Many studies have been concerned with android apps and identifying malicious applications and discovering the patterns of malware samples. Consequently, malware detection of the Android platform becomes a topical matter for many researchers (Su et al., 2020), since the Android platform dominates the market of mobile devices (Peynirci et al., 2020; Garg & Baliyan, 2021). As stated by (Liu et al., 2020), around 86.6% of smartphone devices sold in 2019 were based on the Android operating system, and by the end of the 2020 year will be more than 2.8 million applications distributed to the official store of Android applications (Google Play). That fast growth of android apps is due to the open-source of the Android platform and its ability for invoking third-party code (Liu et al., 2020). The design of the android operating system security model is based on application-oriented mandatory access control and sandboxing that restricts access to local resources by using the permission constraints (Arora et al., 2019). Permission is used as a type of secure layer in the android platform to protect the privacy and data of the user. For instance, applications request permissions to access certain sensitive information of a user such as text messages and contacts, as well as various system components such as the camera, microphone, and network. That permissions could be requested at installation and run time for updated versions of the android operating system. However, they could be abused by attackers and threaten user privacy (Arora et al., 2019). Several reported works have been conducted to handle the malicious apps of the android platform based on machine learning techniques due to their ability in keeping cope with malware development (Ucci et al., 2019; Gibert et al., 2020). The approaches of detecting and classifying android malware using machine learning

methods are grouped into three types based on the features extracted; static approach in which attribute variables are obtained before execution of the application, while the features in the dynamic method are extracted after execution of the application. The hybrid approach includes both static and dynamic features (Yan & Yan, 2018; Gibert et al., 2020; Garg & Baliyan, 2021). Classifying Android applications to malicious and non-malicious using permissions is more popular and effective than other features (Yuan et al., 2020). For instance, classifying android apps based on static permissions is less expensive because the apps are not executed (Milosevic et al., 2017). However, classifying Android apps based on permissions extracted at run time is more efficient but it consumes much time (Milosevic et al., 2017). Hybrid features are more comprehensive in classifying Android apps. In the domain of machine learning along with the era of big data and the tremendous explosion of mobile apps, detecting and classifying malware apps from non-malware apps among those massive data becomes more challenging (Zhang et al., 2019). Therefore, finding the discriminative and relevant permission features that distinguish malware apps from non-malware apps becomes the interest of many researchers. Consequently, numerous researchers have been adopted feature selection and classification methods to detect and classify android malicious apps from non-malicious apps. For instance, (Aminordin et al., 2018; Singh et al., 2019; Alazab, 2020; Faris et al., 2020; Mahindru, 2020; Sangal, 2021) used feature selection approaches to obtain the most discriminative features subset from the original features that are capable to classify malware apps from non-malware apps and improve the classification performance with less computational cost overhead by using a small number of significant features. However, designing a promising and robust approach that deals with a large dataset with large features is the concern of this research study.

1.2 Problem Background

With the development of smartphones and the combination of their applications currently, people depend mostly on their mobile devices for accessing the

Internet, online banking, shopping, emails, social media access, online learning, and mobile-health, etc. (Ghasempour et al., 2020; Jaenke et al., 2021; López-Moranchel et al., 2021). Those facilities have led to the development and distribution of free Mobile apps for Android platforms in particular on the market which attracts many users to download them without awareness. As a result, users attend to store their sensitive personal information such as passwords, mail, and bank account details on their mobile devices along with using those apps. That trend has led to the dramatic growth of the number of malware threats (Kouliaridis et al., 2020; Feng et al., 2021). As reported by McAfee, there are some hidden apps that are able to hide their existence after installation and disturb user victims with an irritating advertisement, representing the most active threat to mobile users (Kouliaridis et al., 2020). That type of app is designed with encrypted code and is able to spread throughout the program (Kouliaridis et al., 2020). Mobile malware is malicious software designed to exploit a mobile device without the owner's knowledge with bad intentions. It can be categorized as a virus, trojan, worm, or botnet (Appice et al., 2020; Alazab et al., 2020). A malicious app could steal the sensitive information of a user, make phone calls and send text messages. For example, svpeng malware could steal bank IDs and credit card numbers by tricking the user to enter information in the window as shown in Figure 1.1 below, and breaking user privacy (Maiorca et al., 2015).

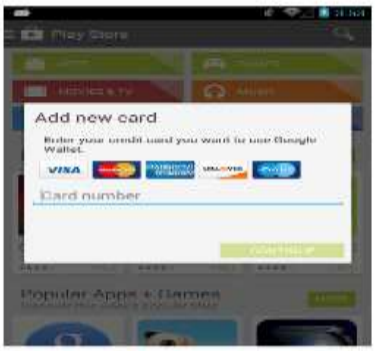

Figure 1.1 The svpeng malware with a phishing window asking credit card details

In addition, the design of the android operating system security model is based on application-oriented mandatory access control and sandboxing that restricts the

access to local resources by using the permission constraints (Arora et al., 2019). For example, each application must assign a unique User ID and a set of permissions at the installation time of the app to provide a limitation for accessing different features as a type of protection (Bhandari et al., 2017; Kumar et al., 2018; Wang et al., 2019). However, the requested permissions at installation time were implemented in earlier versions of Android (5.1 and below) as an ask-on-install (AOI) policy that requires users to grant permissions at installation time while the new updated permissions mechanism in Android 6.0 rely on the ask-on-first-use (AOFU) policy that requires user authorization to be provoked at runtime for user decision (Gao et al., 2020). Even though permissions provide a secure layer to users by restricting access to local resources, they can be exploited by the attackers to threaten user privacy. That exploitation could be done via camera, SMS, call, audio, and image or location exploitation by attacking the system call, permission, or API inside the Android smartphone (Saudi et al., 2017; Alenezi & Almomani, 2018; Ghasempour et al., 2020; Garg & Baliyan, 2021). When granting some permissions, they could compromise user privacy and cause a financial cost for a mobile user such as (send_sms, receive_sms, read_contacts, etc.). For instance, (send_sms) permission enables the app to send text messages without user knowledge and an attacker might abuse it by communicating with a third party or sending text messages to specific numbers, without user consent. Moreover, malicious apps may cost users money by sending messages without user confirmation and could cause financial loss with unexpected charges (Deypir, 2019). In addition, there are some permissions that can cause integrity threats to the operating system (OS), files, and the physical device (Dini et al., 2018) such as (change_wifi_state, install_packages, write_external_storage). For example, the permission (write_external_storage) allows an app to write or modify the external storage of a mobile device and can damage the memory of the device by filling it and controlling or modifying files constantly. An example of malware using this permission is (Moghava) malware that can destroy all the photos of the user gallery by loading an advertising image instead (Dini et al., 2018). Consequently, many reported works intend to explore the permissions mechanism and analyze them to find the patterns of permissions that are associated with malicious apps by utilizing machine learning techniques. These approaches of machine learning can be categorized based on features extracted to static analysis in which features are extracted at installation

time and dynamic analysis where features are extracted at run time and hybrid analysis that includes both static & dynamic features (Wang et al., 2019). For example, Wang et al. (2014), analysis the risk of permissions that are requested at installation time (static features) based on three phases. Firstly, they analyzed the risk of the individual permission features followed by studying the risk of a combination of permission features to investigate the critical permissions that are related to suspicious apps. The study done by Talha et al. (2015) analyzed android permissions (static features) by building a system called APK Auditor on a server based on assigning scores to the requested permissions by apps (Talha et al., 2015). However, they explore only the permissions that are requested at installation time (static permission features). Jin et al. (2018) designed the (SIGPID) method to investigate the importance of requested permissions in differentiating between malware and non-malware apps. They used ranking methods with negative rate and association rules to select the most important permission features. And they used different classifier algorithms to classify suspicious apps from non-suspicious ones.

Doğru & Önder (2020) designed a system called AppPerm Analyzer to extract the permissions of analysed applications and computes risk scores of these permissions and they created permission groups according to their usage rates in suspicious and non-suspicious applications and calculate the total risk score of that analyzed applications. The permissions that are explored in their study represent the static permissions obtained at installation time.

Ikwuegbu (2020) proved that mobile application attributes such as permissions, ratings, and number of installations have an impact on identifying the violence of mobile applications. They utilized different machine learning classifiers such as (K-NN and RF) to evaluate the security risk of mobile apps based on the mentioned attribute features.

Most of the mentioned previous studies concentrate mostly on studying the permissions at installation time. These permissions are more important to be explored

because a security model that would inform users about permission settings only once during application installation has been considered insufficient. Moreover, many users are careless about these permissions and that leads to security and privacy threats (Alepis & Patsakis, 2017; Arora et al., 2019). However, the updated version of the android platform requests permissions during runtime to restrict access to some respective resources of the mobile devices. These permissions are categorized as dangerous permissions based on android website developers (Alepis & Patsakis, 2017) & (Mahindru & Singh, 2017) and they are more critical because they describe how sensitive information of the user is handled by the applications. Therefore, this research study explores the permissions extracted at installation time (static permission features) along with hybrid permissions that include static and (dynamic permissions that are extracted at run time; which means permissions extracted after executing the apps). For example, the study conducted by (Mahindru & Singh, 2017) used hybrid permission features (static and dynamic permission features) to detect android malware by utilizing machine learning classifiers. In this research study, the new version of the dataset used in the work of (Mahindru & Singh, 2017) has been used along with other datasets as described in chapter 3. That dataset has 25458 samples (8643 malware apps & 16815 benign apps) with 173 permission features.

Since many research studies investigate different feature variables along with permission, the motivation of this study is to explore only permissions as features for the dataset by utilizing machine learning classifiers to build a classifier model. Different machine learning classifiers are utilized to find out which classifier model is suitable for Android malware classification. Ensemble classifier algorithms such as random forest and extra tree are employed in this study to investigate the effectiveness of these classifiers on classification performance as well. However, with the era of big data and the explosive growth of mobile apps distribution currently along with the rapid growth of malware apps (Wang et al., 2019; Doğru & Önder, 2020), the interpretation of a dataset becomes more difficult and visualization of that high dimensional dataset with many redundant and irrelevant features becomes also more challenging and not clear (Wang et al., 2019). Furthermore, the number of permissions requested by android apps is growing exceedingly with the growth of the android apps

as well; could be more than 300 permissions, as reported by (Alswaina & Elleithy, 2018) and malicious apps request many unnecessary permissions (Arora et al., 2019; Qiu et al., 2019; Shrivastava et al., 2019). And these growing numbers of that permission features will make it complex and would misguide classifiers by over-fitting of data. Thus, analysing and classifying apps with that huge permission features comprising of non-relevant and redundant permission features will increase the computational overhead and also will decay the classification performance (Xu et al., 2017; Wang et al., 2019; Salah et al., 2020). Further, learning with a large dataset with whole features will cause data redundancy and increase data noise which leads to degeneration of classification, increasing the complexity of understanding the model and raising the computational cost overhead (Feng et al., 2018; Yuan et al., 2020). In addition, analysing permissions and extracting useful knowledge and patterns from such a huge number of applications and noisy data presents a challenging task since applications request many different permissions and will be there more redundant or irrelevant permission features. Additionally, non-relevant permission features are not related to class label and will affect the learning process and redundant permissions features are useless and do not add anything to the target concept which leads to lower the classification performance and increase the complexity overhead (Dash & Liu, 1997; Wang et al., 2019). Therefore, extracting a small number of optimal and relevant permission features subset in classifying Android apps instead of using all features becomes necessary and desirable to enhance the learning accuracy for classification, lower computational cost, and decrease memory usage (Goecks et al., 2020).

Feature selection is a method of dimensionality reduction that deals with the issues of high-dimensional datasets and reducing the number of features or variables by selecting small subsets of features that are correlated to the target object with preserving the information of the dataset (Dash & Liu, 1997; Aggarwal et al., 2014). Feature selection is widely employed in classifying and detecting android malware to select a subset of features from the original feature set without any transformation and maintain the physical meanings of the original features. Several reported works such as (Gillies, 2015; Vinod et al., 2019; Zebari et al., 2020) have demonstrated the effect of feature selection methods in minimizing redundancy and maximizing relevance to

the target such as the class labels in classification. The studies conducted by (La & Mar 2018; Salah et al., 2020; Arvind Mahindru, 2020; Sangal, 2021) pointed out that using a huge number of features leads to an increase in the time of learning, raises in memory usage, and decays the classification performance; and feature selection approaches represent one of the strategies used to handle that issues. Generally, there are three kinds of feature selection methods: filter, wrapper, and embedded methods. Filter methods select features based on their relevance to the class label without involving any machine learning algorithm, such as information gain (IG). Wrapper methods identify the best features based on the optimal performance of learning algorithms, such as particle swarm optimization (PSO). Embedded methods incorporate feature selection as a step of the machine learning process, such as decision tree (DT), L -support vector machine (SVM), and sparse logistic regression (Xu et al., 2017).

The study done by (Altaher, 2016) used the Information Gain algorithm (IG) to select the most important permission features and then used three classifiers algorithms (Naïve Bayesian, Random Forest, and J48) to classify android apps based on that reduced permission features. However, they used a small dataset of (100 malware and 100 non-malware apps) and they explored only permissions extracted at installation time. In addition, IG computes an IG value for every feature. The larger the IG value is, the more information the feature contains, and the more important the feature is. IG does not consider the influence of the feature on the classifiers.

This study attempts to answer the questions of how permission features contribute in differentiating between malware and non-malware and how to explore the patterns of the most important permission features for a dataset using the extra tree algorithm. This study is based on the work of (Alswaina & Elleithy, 2018). However, they have examined the permissions that are requested at installation time (static permissions) to classify the applications into their proper malware families by utilizing the extra tree algorithm to reduce the number of permission features from 59 to 42 (by 0.28%) while this research study focuses on classifying apps to malicious and non-malicious based on static and hybrid permissions (static and dynamic permissions) as

well. Also, this study is different from the study of Alswaina & Elleithy (2018) by selecting a small number of the best features (top 5, 10, and 20 features) from the list of most reduced important features by utilizing the largest (n) function as explained in the next chapters.

Many research studies have leveraged the extra-tree algorithm classifier to compute the importance score of each attribute based on the feature selection approach such as (Pektaş & Acarman, 2018; Suarez-tangil et al., 2017; Vinod et al., 2019). They also proved its robustness and its computational efficiency in selecting subsets of optimal and discriminative features that contribute towards enhancing the classification performance. Extra tree algorithm utilized feature importance property which is an inbuilt class with an extra tree model that extracts features based on assigning scores to input features of the predictive model by computing (Gini Index) in the decision of feature spilled of the data to determine the features with the relative importance towards the class label. And that helps in understanding the dataset better and since that features are scored based on the extra tree classifier algorithm in which the dataset has been fitted, that leads to a better understanding of the model by determining which feature attributes that are most and least prominent to the classifier model. While information gain approach selects features based on their information towards the class target regardless of which classifier models or ML algorithms are used. In addition, selecting a less number of important features instead of using all features helps in reducing the time of processing, helps in better learning performance, and helps in lowering computational cost overhead (Das & Das, 2017; Kumar et al., 2018; Bruno, 2019; Haq et al., 2019). However, representing the discriminative features extracted based on feature selection methods with (Boolean values) instead of weighted values in the learning model will assume that all features are equally important but it is not always the case since some features have a stronger influence on malware classification while others have less impact (Xu et al., 2017; Cai et al., 2021). For instance, the SMS permission features are often used in malicious applications but are used less often in non-malicious applications. Therefore, SMS-related information has a powerful impact on Android app classification (Cai et al., 2021). Thus, computing feature weight will help in determining the features with high

influence in distinguishing between malware and non-malicious apps. Moreover, representing features with weight values helps in improving classification performance as reported by many research works such as (Buckley, 1993; Unal et al., 2014; Alswaina & Elleithy, 2018; Kumar et al., 2018; Vinod et al., 2019). Additionally, many research works adopted a single feature selection method to obtain the most critical features. However, some single approaches of feature selection methods could have some limitations but some other techniques might not have those limitations. Therefore, combining two approaches of feature selection methods will help in getting rid of the limitations of one method by making use of each other's strengths (Bhattacharya et al., 2018; Haq et al., 2019). Therefore, the aim of this research is to propose an approach that combines two methods of feature selection techniques to obtain more effective results. The Information Gain method (IG) is combined with the ensemble-based extra tree feature selection algorithm to compute the weights of the selected features. IG selects features based on their relevance to the class label with regardless of using any classifier model. Thus, using only values of IG as feature weights maybe not be appropriate to be used for each feature and not helpful in improving the accuracy of the final classifier for malware detection (Cai, Li, & Xiong, 2021). Therefore, the extra tree algorithm is used in this study to compute the weights of features to get the most robust features by utilizing the feature importance property that comes with the algorithm. Several studies have been adopted the approach term frequency-inverse document frequency (TF-IDF) for assigning weights to feature variables to differentiate between their corresponding importance such as done in the works of (Fan et al., 2018; Kural et al., 2018; Sahal et al., 2018; Su et al., 2018; Jannat et al., 2019; Deepa et al., 2019; Darabian et al., 2020; Dharmalingam & Palanisamy, 2020; Salah et al., 2020; Yuan et al., 2020). Nevertheless, using (TF-IDF) methods for feature weighting is computationally expensive because many permission features occur many times, and computing the frequency statistics for all that features will be more expensive computationally. For instance, SMS-related permission features may appear several times in one application's source code but may have a low frequency overall in the source code file (Xu et al., 2017). Hence, using the extra tree algorithm to compute weight is more efficient and saves time because the whole procedure is the

same in constructing the decision trees in which features are selected randomly based on Gini Index mathematical criteria (Moosmann et al., 2007).

1.3 Problem Statement

With the revolution of big data and the huge distribution of android mobile apps and malicious apps, android permissions play important role in differentiating between malware and non-malware apps as pointed out by many studies, however, the number of permission features is growing exceedingly with the growth of mobile apps. These growing numbers of permission features will increase the computational cost overhead and would lead to decreasing classification performance, so using a small number of significant permission features instead of using all permission features will decrease complexity and enhance the classification performance of classifying android apps.

Based on the problem background presented previously, the problem statement for this study is:

Given a binary class dataset of Android applications with malware and non–malware samples and permission feature attributes, the challenge is to address the issues of reducing the irrelevant and redundant permission features that make the representation of the dataset difficult and decay the performance of the classification model. Ensemble-based extra tree algorithm is robust in extracting the most discriminative and relevant features that are related to the class target with preserving information of the dataset. Therefore, the weighted ensemble-based extra tree integrated with the information gain-based feature selection-based scheme is proposed to explore, analyse and obtain the most prominent and highly influential weighted permission features that contribute towards the target variable with less number of features.

This thesis explores the above-stated research problem with the help of the following research questions;

i. How does the size of features or attributes of the dataset affect on the classification performance?

ii. What is the impact of feature selection methods on classification performance?

iii. How to explore the patterns of the most important features for the dataset?

iv. How to determine the most optimum feature subset using ensemble-based extra tree feature selection for various distributions of datasets?

v. How to evaluate the significant permissions in classifying android apps?

vi. How does the representation of features affect the classification performance?

vii. How to compute the weight of the attribute features in the dataset?

viii. How to test and evaluate the weighted features subset in the classification task?

1.4 Aim of the Research

The aim of this study is to investigate the effect of permission features in classifying android apps and to obtain the small number of discriminative weighted permission features that classify android malware apps from non-malware apps instead of using all available permission features set by using the integrated Information Gain (IG) with ensemble-based extra tree feature selection algorithm and analysing that significant permission features would help in reducing the complexity overhead as well as improving the classification performance.

1.5 Objectives of the Research

This research embarks on the following objectives:

1. To investigate the effect of permission features with different classifier models in classifying Android applications.

2. To propose and design an ensemble-based extra tree feature selection scheme (FS-EX) to obtain highly discriminant permission feature subsets for classifying Android applications.

3. To propose and design an integrated Information Gain with ensemble-based extra tree feature selection (FS-IGEX) for improved weighted permission features in classifying Android applications.

1.6 Research Scope

The scope of the research is limited to the following:

i. The research focuses on investigating android permission features extracted at installation and run time that helps in classifying android apps as malicious or non-malicious apps.

ii. Five different datasets of various data distribution and features size with real samples of malware and benign namely MO, Machine Learning 1, Machine Learning 2, Kaggle, and Hybrid dataset collected from different resources as explained in chapter 3 are used to validate the performance of the proposed methods. The permission features for Machine Learning 1 and Machine Learning 2 datasets are kept while the other features are excluded.

iii. The proposed methods are used only to address the classification of binary class problems.

iv. The performance evaluation is carried out to evaluate the classification performance between the proposed method (FS-EX) and using the full permission features (FP) and between the binary representation and weighted representation of permission features (PB & PW) of the proposed method (FS-IGEX) and between the weighted representation of permission features (PW) approach and using the full permission features (FP) approach.

1.7 Contributions of the Research

This research work has accomplished the following novel contributions to the existing body of knowledge:

i. Created MO dataset with static permission features that are extracted before executing the apps (43 permissions with 260 samples) which consists balance of real-world samples of malware and non-malware collected during the period (2016-2017).

ii. Investigating and analyzing the static permissions that are requested at installation time and also permissions that are requested at run time by exploring the Hybrid dataset with the proposed methods (FS-EX & FS-IGEX). That dataset has (99 static permissions and 74 dynamic permissions).

iii. Proposing Ensemble-Based Extra Tree Feature Selection Algorithm (FS-EX): Ensemble-Based Extra Tree Feature Selection Algorithm (FS-EX) method tends to identify the subset of the most important discriminative permission features in classifying malicious android apps and non-malicious apps and visualizing them by utilizing feature importance property class that is built in the model classifier (extra tree). Its performance has been validated with five different data sets with varying permission features distribution and different distribution of samples. The comparison studies demonstrated that the less number of permission feature subsets obtained with the proposed method (FS-EX) achieve the same or close performance results compared to using the whole permission features (FP). The advantage of the proposed method demonstrates that using a small number of significant permission features instead of employing all available permission features (FP) will lead to decreasing complexity and getting better or close performance measures results compared to all permission features since the feature importance selects the reduced prominent features that are relevant to the class target. This is reflected in the results of Duncan's Multiple Range Statistical Test (DRMT) presented in Chapter 5.

iv. Proposing (FS-IGEX) approach by integrating Information Gain with an Extra Tree algorithm to assign weights to important features of the datasets used in this study. FS-IGEX extracts the top 5, 10, and 20 features firstly by using the Information Gain (IG) method and then the ensemble-based extra tree features selection algorithm is applied on the updated feature subsets of 5, 10, and 20 features obtained using (IG). The features' weights were calculated using the feature importance property class that is built within the extra tree model classifier. The performance of the proposed (FS-IGEX) method has been validated with five different data sets same as done with the (FS-EX) method. The advantage of the proposed (FS-IGEX) method demonstrates that assigning weights to features helps in determining the importance of features in the dataset that have a higher influence on the class label; for example, the features with the higher weights have the highest impact on the class target while the features with the lowest weight have the lowest impact on the class target. Additionally, learning with a small subset of weighted features results in better classification performance. This is demonstrated by the results of Duncan's Multiple Range Statistical Test (DRMT) presented in Chapter 6.

v. A comparison study on the effect of representing features in Boolean values and weightage values on the classification performance. Such comparison study has been carried out to learn how the representation of features in learning task plays role in enhancing the model performance, for instance, features with weightage values have led to improving the classification performance with some classifiers and obtained explanation about features that have a high and low impact on the target object. This is reflected in the results of Duncan's Multiple Range Statistical Test (DRMT) presented in Chapter 6.

vi. Identifying and analyzing the significant and risky permission features

that contribute in classifying android apps based on the top-ranked features selected by the proposing methods (FS-EX) and (FS-IGEX). That features have been visualized and illustrated in detail in section 5.3 in chapter 5 and in section 6.2.4 in chapter 6.

vii. Evaluating the performance of the proposed methods (FS-EX) and (FS-IGEX) by classifying different datasets with the different number of sample sizes and attribute features.

viii. Reducing the feature space size by using small datasets, which resulted in a small feature space such as using MO and Kaggle datasets. Reducing the feature space size by considering a limited number of feature categories, e.g., Android permissions only. Using ranking and feature selection methods to obtain the most important features.

1.8 Significance of the Research

The significance of this research could be described in two main categories, computational and domain applications developers. From a computational view, the proposed methods (FS-EX) and (FS-IGEX) intend to speed up the learning process and improve the classification performance task by identifying the small number of relevant, discriminative, and informative feature subsets that contribute toward classifying apps as malware and non-malware rather than using all permission features. Also, choosing and selecting fewer features will help in decreasing time-consuming and expensive computing costs instead of using the whole features. Moreover, using the most essential and discriminating subset of features instead of using all features will give insight about the dataset since it is easy to visualize the dataset with small subsets of features and leads to better model interpretability as well. Additionally, it is desirable because the classification performance increased with some classifiers and

computational overhead is reduced with less number of features. And it is significantly important if the proposed methods used in other domains, for example, in investigating disease; by employing the most highly discriminative, informative, and correlated features towards the class target will help in inspecting and finding the relevant feature patterns to a specific disease and which have the highest impact in diagnosing diseases and lead to better learning performance.

From the view of domain applications android developer, the proposed method analyses the permission patterns by listing and visualizing the most important of permissions that are related to the class label and determining the degree of risk that features held will give more insight about dangerous permissions and will help the developer to avoid those permissions and also will increase the awareness of users about the danger of specific permissions and increase their awareness of the necessity of avoiding granting permissions without knowledge.

1.9 Organization of the Thesis

This thesis study is organized into seven chapters. Each chapter presents its contents as follows:

i. Chapter 1 – Introduction: This chapter presents the motivations of the research and the justification of the proposed methods. The research problem statement, objectives, and significance are defined. In addition, the potential contribution of the research has also been outlined.

ii. Chapter 2 – Literature Review: This chapter summarized the previous works in the studies of android applications classification based on

analyzing permissions, the current trend, and future directions. The selected techniques used in the proposed methods are also described.

iii. Chapter 3 – Methodology: This chapter explains the architectural design of the proposed methods, along with its operational and implementation framework. The performance metrics and data sets used for classifier evaluation are also presented.

iv. Chapter 4 – The effect of using permission features in classifying android apps: This chapter addresses the impact of permission features in classifying android apps and what suitable classifier model helps in improving the classification performance.

v. Chapter 5 – Ensemble-Based Extra Tree Classifier Feature Selection to Explore Android Permissions (FS-EX): This chapter purposes an extra tree classifier as a feature selection-based method to explore the android permissions and select the optimal permission features subset that is related to the target variable.

vi. Chapter 6 – Information Gain Integrated with Ensemble-Based Extra Tree Classifier Feature Selection Classifier: This chapter proposes the (FS-IGEX) method based on weighting permission features to obtain the highly influential permission features in classifying android apps and to improve the classification quality.

vii. Chapter 7 – Conclusion: This chapter draws general conclusions from the results, summarizes the important findings of the study, and presents the research contributions with some suggested future works.

REFERENCES

Abawajy, J., Darem, A., & Alhashmi, A. A. (2021). Feature subset selection for malware detection in smart iot platforms. Sensors (Switzerland), 21(4), 1–19. https://doi.org/10.3390/s21041374.

Abbasi, F. U. (2013). Detection and classification of malicious network streams in honeynets: a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science at Massey University, Palmerston North, New Zealand (Doctoral dissertation, Massey University).

Abdulla, S., & Altaher, A. (2015). Intelligent approach for android malware detection. KSII Transactions on Internet and Information Systems, 9(8), 2964–2983. https://doi.org/10.3837/tiis.2015.08.012.

Abdullah, Z., & Saudi, M. M. (2018). RAPID-Risk assessment of android permission and application programming interface (API) call for android botnet. International Journal of Engineering and Technology(UAE), 7(4). https://doi.org/10.14419/ijet.v7i4.15.21370.

Abubaker, H., Shamsuddin, S. M., & Ali, A. (2018). Analytics on malicious android applications. International Journal of Advances in Soft Computing and Its Applications, 10(1), 106–118.

Aggarwal, C. C., Kong, X., Gu, Q., Han, J., & Yu, P. S. (2014). Active learning: A survey. Data Classification: Algorithms and Applications, 571–605. https://doi.org/10.1201/b17320.

Alam, M. S. (2016*). An Intelligent Multi-Agent based Detection Framework for Classification of Android Malware*, PhD Thesis, The University of British Columbia, Vancouver.

Alam, S., Alharbi, S. A., & Yildirim, S. (2020). Mining nested flow of dominant APIs for detecting android malware. Computer Networks, 167, 107026. https://doi.org/10.1016/j.comnet.2019.107026.

Alazab, M. (2020). Automated malware detection in mobile app stores based on robust feature generation. Electronics (Switzerland), 9(3). https://doi.org/10.3390/electronics9030435.

Alazab, M., Alazab, M., Shalaginov, A., Mesleh, A., & Awajan, A. (2020). Intelligent mobile malware detection using permission requests and API calls. Future

Generation Computer Systems, 107(September), 509–521. https://doi.org/10.1016/j.future.2020.02.002.

Alenezi, M., & Almomani, I. (2018). Abusing Android permissions: A security perspective. 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2017, 2018–Janua, 1–6. https://doi.org/10.1109/AEECT.2017.8257772.

Alepis, E., & Patsakis, C. (2017). Hey doc, is this normal? Exploring android permissions in the post marshmallow Era. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10662 LNCS(December), 53–73. https://doi.org/10.1007/978-3-319-71501-8_4.

Al Jutail, M., Al-Akhras, M., & Albesher, A. (2019). Associated Risks in Mobile Applications Permissions. Journal of Information Security, 10(2), 69–90. https://doi.org/10.4236/jis.2019.102004.

Al-Kaaf, H., Ali, A., Shamsuddin, S., & Hassan, S. (2020). Feature selection for malicious android applications using Symmetrical Uncert Attribute Eval method. IOP Conference Series: Materials Science and Engineering, 884(1). https://doi.org/10.1088/1757-899X/884/1/012060.

Al-shahib, A., England, P. H., Breitling, R., & Gilbert, D. R. (2005). Feature Selection and the Class Imbalance Problem in Predicting Protein Function from Sequence, (February). https://doi.org/10.2165/00822942-200594030-00004

Alshehri, A. (2019). PUREDroid : Permission Usage and Risk Estimation for Android Applications, 179–184.

Alshehri, A., Hewins, A., McCulley, M., Alshahrani, H., Fu, H., & Zhu, Y. (2017). Risks behind Device Information Permissions in Android OS. Communications and Network, 9(4), 219–234. https://doi.org/10.4236/cn.2017.94016.

Alshwaheen, T. I., Hau, Y. W., Ass'Ad, N., & Abualsamen, M. M. (2021). A Novel and Reliable Framework of Patient Deterioration Prediction in Intensive Care Unit Based on Long Short-Term Memory-Recurrent Neural Network. IEEE Access, 9, 3894–3918. https://doi.org/10.1109/ACCESS.2020.3047186.

Alsoghyer, S., & Almomani, I. (2019). Ransomware detection system for android applications. Electronics (Switzerland), 8(8), 1–36. https://doi.org/10.3390/electronics8080868.

Alswaina, F., & Elleithy, K. (2018). Android Malware Permission-Based Multi-Class Classification Using Extremely Randomized Trees. IEEE Access, 6, 76217–76227. https://doi.org/10.1109/ACCESS.2018.2883975.

Altaher, A., & Mohammed, O. (2017). Intelligent Hybrid Approach for Android Malware Detection based on Permissions and API Calls. International Journal of Advanced Computer Science and Applications, 8(6), 60–67. https://doi.org/10.14569/ijacsa.2017.080608.

Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep learning based android malware detection using real devices. Computers and Security, 89. https://doi.org/10.1016/j.cose.2019.101663.

Amamra, A., Robert, J. M., Abraham, A., & Talhi, C. (2016). Generative versus discriminative classifiers for android anomaly-based detection system using system calls filtering and abstraction process. Security and Communication Networks, 9(16), 3483–3495. https://doi.org/10.1002/sec.1555.

Aminordin, A., Abdollah, M. F., & Yusof, R. (2018). Android malware classification base on application category using static code analysis. Article in Journal of Theoretical and Applied Information Technology. Retrieved from https://www.amazon.com/mobile-apps/b.

Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. Information (Switzerland), 11(6). https://doi.org/10.3390/info11060332.

Appice, A., Andresini, G., & Malerba, D. (2020). Clustering-Aided Multi-View Classification : A Case Study on Android Malware Detection.

Arora, A., Peddoju, S. K., & Conti, M. (2019). PermPair : Android Malware Detection Using Permission Pairs. IEEE Transactions on Information Forensics and Security, 15, 1968–1982. https://doi.org/10.1109/tifs.2019.2950134.

Arp, D., Spreitzenbarth, M., Malte, H., Gascon, H., & Rieck, K. (2014). Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. Symposium on Network and Distributed System Security (NDSS), (FEBRUARY), 23–26. https://doi.org/10.14722/ndss.2014.23247.

Arshad, S., Shah, M. A., Wahid, A., Mehmood, A., Song, H., & Yu, H. (2018). SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System. IEEE Access, 6, 4321–4339. https://doi.org/10.1109/ACCESS.2018.2792941.

Arslan, R. S., Doğru, İ. A., & Barişçi, N. (2019). Permission-Based Malware Detection System for Android Using Machine Learning Techniques. International Journal of Software Engineering and Knowledge Engineering, 29(1), 43–61. https://doi.org/10.1142/s0218194019500037.

Arvind Mahindru, and A. L. S. (2020). DLDroid: Feature Selection based Malware Detection Framework for Android Apps developed during COVID-19. International Journal on Emerging Technologies, 11(3), 516–525.

Bhandari, S., Jaballah, W. Ben, Jain, V., Laxmi, V., Zemmari, A., Gaur, M. S., Conti, M. (2017). Android inter-app communication threats and detection techniques. Computers and Security, 70, 392–421. https://doi.org/10.1016/j.cose.2017.07.002.

Bhandari, S., Panihar, R., Naval, S., Laxmi, V., Zemmari, A., & Gaur, M. S. (2018). SWORD: Semantic aWare andrOid malwaRe Detector. Journal of Information Security and Applications, 42, 46–56. https://doi.org/10.1016/j.jisa.2018.07.003.

Bhattacharya, A., Tamal, R., & Kuntal, G. (2018). A feature selection technique based on rough set and improvised PSO algorithm (PSORS-FS) for permission based detection of Android malwares. International Journal of Machine Learning and Cybernetics, 0(0), 0. https://doi.org/10.1007/s13042-018-0838-1.

Bruno, L. (2019). Machine Learning Algorithms. Journal of Chemical Information and Modeling (Vol. 53). https://doi.org/10.1017/CBO9781107415324.004

Buckley, C. (1993). The importance of proper weighting methods, (January 1993), 349. https://doi.org/10.3115/1075671.1075753

Cai, L., Li, Y., & Xiong, Z. (2021). JOWMDroid: Android malware detection based on feature weighting with joint optimization of weight-mapping and classifier parameters. Computers and Security, 100, 102086. https://doi.org/10.1016/j.cose.2020.102086.

Camana Acosta, M. R., Ahmed, S., Garcia, C. E., & Koo, I. (2020). Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart

grid networks. IEEE Access, 8(Ml), 19921–19933. https://doi.org/10.1109/ACCESS.2020.2968934.

Camilo, C., & López, U. (2016). Framework for malware analysis in Android, (August). https://doi.org/10.18046/syt.v14i37.2241.

Canfora, G., Mercaldo, F., Medvet, E., & Visaggio, C. A. (2015). Detecting Android malware using sequences of system calls. 3rd International Workshop on Software Development Lifecycle for Mobile, DeMobile 2015 - Proceedings, 13–20. https://doi.org/10.1145/2804345.2804349.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers and Electrical Engineering, 40(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.

Chavan, N., Di Troia, F., & Stamp, M. (2019). A comparative analysis of android malware. ICISSP 2019 - Proceedings of the 5th International Conference on Information Systems Security and Privacy, (November), 664–673. https://doi.org/10.5220/0007701506640673.

Chen, L., Ye, Y., & Bourlai, T. (2017). Adversarial machine learning in malware detection: Arms race between evasion attack and defense. Proceedings - 2017 209 European Intelligence and Security Informatics Conference, EISIC 2017, 2017–Janua, 99–106. https://doi.org/10.1109/EISIC.2017.21.

Chen, T., Mao, Q., Yang, Y., Lv, M., & Zhu, J. (2018). TinyDroid: A lightweight and efficient model for android malware detection and classification. Mobile Information Systems, 2018. https://doi.org/10.1155/2018/4157156.

Damshenas, M., Dehghantanha, A., Choo, K.-K. R., & Mahmud, R. (2015). M0Droid: An Android Behavioral-Based Malware Detection Model. Journal of Information Privacy and Security, 11(3), 141–157. https://doi.org/10.1080/15536548.2015.1073510.

Darabian, H., Dehghantanha, A., Hashemi, S., Taheri, M., Azmoodeh, A., Homayoun, S.,Parizi, R. M. (2020). A multiview learning method for malware threat hunting : windows, IoT and android as case studies.

Das, A., & Das, S. (2017). Feature weighting and selection with a Pareto-optimal trade-off between relevancy and redundancy. Pattern Recognition Letters, 88, 12–19. https://doi.org/10.1016/j.patrec.2017.01.004.

Dash, M., & Liu, H. (1997). Feature selection for classification. Intelligent Data Analysis, 1(3), 131–156. https://doi.org/10.3233/IDA-1997-1302.

Deepa, K., Radhamani, G., & Vinod, P. (2015). Investigation of feature selection methods for android malware analysis. Procedia Computer Science, 46(Icict 2014), 841–848. https://doi.org/10.1016/j.procs.2015.02.153.

Deypir, M. (2019). Entropy-based security risk measurement for Android mobile applications. Soft Computing, 23(16), 7303–7319. https://doi.org/10.1007/s00500-018-3377-5.

Dharmalingam, V. P., & Palanisamy, V. (2020). A novel permission ranking system for android malware detection—the permission grader. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-020-01957-5.

Dini, G., Martinelli, F., Matteucci, I., Petrocchi, M., Saracino, A., & Sgandurra, D. (2018). Risk analysis of Android applications: A user-centric solution. Future Generation Computer Systems, 80, 505–518. https://doi.org/10.1016/j.future.2016.05.035.

Doğru, I. A., & Önder, M. (2020). AppPerm Analyzer: Malware Detection System Based on Android Permissions and Permission Groups. International Journal of Software Engineering and Knowledge Engineering, 30(3), 427–450. https://doi.org/10.1142/S0218194020500175.

Dunham, K., Hartman, S., Morales, J. A., Quintans, M., & Strazzere, T. (2014). Android malware and analysis.

Effrosynidis, D., & Arampatzis, A. (2021). An evaluation of feature selection methods for environmental data. Ecological Informatics, 61(April 2020), 101224. https://doi.org/10.1016/j.ecoinf.2021.101224.

Fahim, U. H. (2013). Detection and classification of malicious network streams in honeynets: a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science at Massey University, Palmerston North, New Zealand (Doctoral dissertation, Massey University).

Fan, M., Liu, J., Luo, X., Chen, K., Tian, Z., Zheng, Q., & Liu, T. (2018). Android malware familial classification and representative sample selection via frequent subgraph analysis. IEEE Transactions on Information Forensics and Security, 13(8), 1890–1905. https://doi.org/10.1109/TIFS.2018.2806891.

Fang, Q., Yang, X., & Ji, C. (2019). A Hybrid Detection Method for Android Malware. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 2127-2132). IEEE. https://doi.org/10.1109/ITNEC.2019.8729017.

Fang, Y., Gao, Y., Jing, F., & Zhang, L. (2020). Android Malware Familial Classification Based on DEX File Section Features. IEEE Access, 8, 10614–10627. https://doi.org/10.1109/ACCESS.2020.2965646.

Faris, H., Habib, M., Almomani, I., Eshtay, M., & Aljarah, I. (2020). Optimizing extreme learning machines using chains of salps for efficient android ransomware detection. Applied Sciences (Switzerland), 10(11). https://doi.org/10.3390/app10113706.

Faverjon, C., & Berezowski, J. (2018). Choosing the best algorithm for event detection based on the intend application: A conceptual framework for syndromic surveillance. Journal of Biomedical Informatics, 85(July 2017), 126–135. https://doi.org/10.1016/j.jbi.2018.08.001.

Feizollah, A., Anuar, N. B., Salleh, R., Suarez-Tangil, G., & Furnell, S. (2017). AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection. Computers and Security, 65(March), 121–134. https://doi.org/10.1016/j.cose.2016.11.007.

Feizollah, A., Anuar, N. B., Salleh, R., & Wahab, A. W. A. (2015). A review on feature selection in mobile malware detection. Digital Investigation, 13(July), 22–37. https://doi.org/10.1016/j.diin.2015.02.001.

Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011). A survey of mobile malware in the wild. Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices - SPSM '11, 3–14. https://doi.org/10.1145/2046614.2046618.

Feng, P., Ma, J., Sun, C., Xu, X., & Ma, Y. (2018). A novel dynamic android malware detection system with ensemble learning. IEEE Access, 6, 30996–31011. https://doi.org/10.1109/ACCESS.2018.2844349.

Feng, R., Chen, S., Xie, X., Meng, G., Lin, S. W., & Liu, Y. (2021). A Performance-Sensitive Malware Detection System Using Deep Learning on Mobile Devices. IEEE Transactions on Information Forensics and Security, 16, 1563–1578. https://doi.org/10.1109/TIFS.2020.3025436.

Fereidooni, H., Moonsamy, V., Conti, M., & Batina, L. (2014). Efficient Classification of Android Malware in the wild using Robust Static Features. Journal of Network and Computer Applications, 38, 43–53.

Frank, E., Hall, M. A., & Witten, I. H. (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques" Morgan Kaufmann, Fourth Edition.

Frey, B., & Salkind, N. J. (2018). Duncan â€TM s Multiple Range Test.

Gao, H., Guo, C., Huang, D., Hou, X., Wu, Y., Xu, J., Bai, G. (2020). Autonomous Permission Recommendation. IEEE Access, 8, 76580–76594. https://doi.org/10.1109/ACCESS.2020.2967139.

Gao, H., Guo, C., Wu, Y., Dong, N., Hou, X., Xu, S., & Xu, J. (2019). AutoPer: Automatic recommender for runtime-permission in android applications. Proceedings - International Computer Software and Applications Conference, 1(July), 107–116. https://doi.org/10.1109/COMPSAC.2019.00024.

Garg, S., & Baliyan, N. (2021). Android security assessment: A review, taxonomy and research gap study. Computers and Security, 100, 102087. https://doi.org/10.1016/j.cose.2020.102087.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1.

Ghasempour, A., Fazlida, N., & John, O. (2020). Permission Extraction Framework for Android Malware Detection. International Journal of Advanced Computer Science and Applications, 11(11), 463–475. https://doi.org/10.14569/ijacsa.2020.0111159.

Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, *153*(November 2019), 102526. https://doi.org/10.1016/j.jnca.2019.102526.

Gillies, Z. M. H. andDuncan F. (2015). BioMed Research International (J BIOMED BIOTECHNOL). Computational and Mathematical Methods in Medicine, 2015(1), 2–4. Retrieved from http://dx.doi.org/10.1155/2015/.

Guyon, I., Elisseeff, A., & De, A. M. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157–1182.

HaddadPajouh, H., Dehghantanha, A., M. Parizi, R., Aledhari, M., & Karimipour, H. (2019). A survey on internet of things security: Requirements, challenges, and solutions. Internet of Things, (xxxx), 100129. https://doi.org/10.1016/j.iot.2019.100129.

Hamdi, S. J., Omar, N., AL-zebari, A., Merceedi, K. J., Ahmed, A. J., Salim, N. O. M., … Salih, A. A. (2021). A State of Art Survey for Understanding Malware Detection Approaches in Android Operating System. Asian Journal of Research in Computer Science, (September), 44–60. https://doi.org/10.9734/ajrcos/2021/v11i330266.

Haq, A. U., Zhang, D., Peng, H., & Rahman, S. U. (2019). Combining Multiple Feature-Ranking Techniques and Clustering of Variables for Feature Selection. IEEE Access, 7, 151482–151492. https://doi.org/10.1109/ACCESS.2019.2947701.

He, H., & Ma, Y. (2013). He - Imbalanced Learning - 2014.

Huan Liu. (1998). Feature Selection for Knowledge Discovery and Data Mining.

Huang, Z., Yang, C., Zhou, X., & Huang, T. (2019). A Hybrid Feature Selection Method Based on Binary State Transition Algorithm and ReliefF. IEEE Journal of Biomedical and Health Informatics, 23(5), 1888–1898. https://doi.org/10.1109/JBHI.2018.2872811.

Hussain, S. J., Ahmed, U., Liaquat, H., Mir, S., Jhanjhi, N. Z., & Humayun, M. (2019). IMIAD: Intelligent malware identification for android platform. 2019 International Conference on Computer and Information Sciences, ICCIS 2019, 1–6. https://doi.org/10.1109/ICCISci.2019.8716471.

Idrees, F., Rajarajan, M., Conti, M., Chen, T. M., & Rahulamathavan, Y. (2017). PIndroid: A novel Android malware detection system using ensemble learning methods. Computers & Security, 68, 36–46. https://doi.org/10.1016/j.cose.2017.03.011.

Ilham, S., Abderrahim, G., & Abdelhakim, B. A. (2018). Permission based malware detection in android devices, 1–6. https://doi.org/10.1145/3286606.3286860

Ikwuegbu, C. C. (2020). Models for Risk assessment of Mobile Applications, (May).

Jaenke, R., Van Den Boogaard, C., McMahon, E., & Brimblecombe, J. (2021). Development and pilot of a tool to measure the healthiness of the in-store food

environment. Public Health Nutrition, 24(2), 243–252. https://doi.org/10.1017/S1368980020002025.

Janaka Senanayake, H. K. and M. O. A.-K. (2020). Android Malware Detection using Machine Learning. International Journal of Recent Technology and Engineering, 8(2S12), 65–70. https://doi.org/10.35940/ijrte.b1011.0982s1219.

Jannat, U. S., Hasnayeen, S. M., Bashar Shuhan, M. K., & Ferdous, M. S. (2019). Analysis and Detection of Malware in Android Applications Using Machine Learning. 2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019, (January). https://doi.org/10.1109/ECACE.2019.8679493.

Jin Li, Lichao Sun, Qiben Yan, Zhiqiang Li, Witawas Srisa-an, and H. Y. (2018). Significant Permission Identification for Machine-Learning-Based Android Malware Detection. IEEE Transactions on Industrial Informatics, 14(7), 3216–3225. https://doi.org/10.1109/tii.2017.2789219.

Kabakus, A. T. (2019). What static analysis can utmost offer for android malware detection. Information Technology and Control, 48(2), 235–249. https://doi.org/10.5755/j01.itc.48.2.21457.

Kapratwar, A., Di Troia, F., & Stamp, M. (2017). Static and Dynamic Analysis of Android Malware, (January), 653–662. https://doi.org/10.5220/0006256706530662.

Khaire, U. M., & Dhanalakshmi, R. (2019). Stability of feature selection algorithm: A review. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2019.06.012.

Kouliaridis, V., & Kambourakis, G. (2021). A comprehensive survey on machine learning techniques for android malware detection. Information (Switzerland), 12(5). https://doi.org/10.3390/info12050185.

Kouliaridis, V., Kambourakis, G., Geneiatakis, D., & Potha, N. (2020). Two anatomists are better than one-Dual-level android malware detection. Symmetry, 12(7). https://doi.org/10.3390/sym12071128.

Kumar, A., Kuppusamy, K. S., & Aghila, G. (2018). FAMOUS: Forensic Analysis of MObile devices Using Scoring of application permissions. Future Generation Computer Systems, 83, 158–172. https://doi.org/10.1016/j.future.2018.02.001.

Kumar, R., Zhang, X., Khan, R., & Sharif, A. (2019). Research on Data Mining of Permission-Induced Risk for Android IoT Devices. Applied Sciences, 9(2), 277. https://doi.org/10.3390/app9020277.

Kumar, R., Zhang, X., Wang, W., Kumar, J. A. Y., & Sharif, A. (2020). A Multimodal Malware Detection Technique for Android IoT Devices Using Various Features.

Kumar, S., Viinikainen, A., & Hamalainen, T. (2018). A network-based framework for mobile threat detection. Proceedings - 2018 1st International Conference on Data Intelligence and Security, ICDIS 2018, 227–233. https://doi.org/10.1109/ICDIS.2018.00044.

Kumar, V. (2014). Feature Selection: A literature Review. The Smart Computing Review, 4(3). https://doi.org/10.6029/smartcr.2014.03.007.

Kural, O. E., Akleylek, S., & Kiliç, E. (2018). New Results on Permission Based Static Analysis for Android Malware. https://doi.org/10.1109/ISDFS.2018.8355377.

La, C., & Mar, K. (2018). Permission-based Feature Selection for Android Malware Detection and Analysis. International Journal of Computer Applications, 181(19), 29–39. https://doi.org/10.5120/ijca2018917902.

Le, Q., Boydell, O., Namee, B. Mac, & Scanlon, M. (2018). Deep learning at the shallow end: Malware classification for non-domain experts. Digital Investigation, 26(April), S118–S126. https://doi.org/10.1016/j.diin.2018.04.024.

Li, W., Ge, J., & Dai, G. (2016). Detecting Malware for Android Platform: An SVM-Based Approach. Proceedings - 2nd IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2015 - IEEE International Symposium of Smart Cloud, IEEE SSC 2015, 464–469. https://doi.org/10.1109/CSCloud.2015.50.

Lindorfer, M., Neugschwandtner, M., & Platzer, C. (2015). MARVIN : Efficient and Comprehensive Mobile App Classification Through Static and Dynamic Analysis. IEEE 39th Annual Computer Software and Applications Conference (COMPSAC). https://doi.org/10.1109/COMPSAC.2015.103.

Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., & Liu, H. (2020). A Review of Android Malware Detection Approaches Based on Machine Learning. IEEE Access, 8, 124579–124607. https://doi.org/10.1109/ACCESS.2020.3006143.

Liu, Y., Xu, C., Cheung, S. C., & Terragni, V. (2016). Understanding and detecting wake lock misuses for android applications. Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering, 13-18-NaN-2016(January 2018), 396–409. https://doi.org/10.1145/2950290.2950297. Lopez, C. C. U., & Cadavid, A. N. (2016, April). Machine learning classifiers for android malware analysis. In 2016 IEEE Colombian Conference on Communications and Computing (COLCOM) (pp. 1-6). IEEE.

López-Moranchel, I., Alegre, L. M., Maurelos-Castell, P., Pérez, V. P., & Ara, I. (2021). Theoretical aspects for calculating the mobilized load during suspension training through a mobile application. Applied Sciences (Switzerland), 11(1), 1–10. https://doi.org/10.3390/app11010242.

Louppe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees.

Mahdavifar, S. (2020). DeNNeS : deep embedded neural network expert system for detecting cyber attacks. Neural Computing and Applications, 32(18), 14753–14780. https://doi.org/10.1007/s00521-020-04830-w.

Mahindru, A. (2020). MLDroid — framework for Android malware detection using machine learning techniques. Neural Computing and Applications (Vol. 4). Springer London. https://doi.org/10.1007/s00521-020-05309-4.

Mahindru, A., & Sangal, A. L. (2020). DeepDroid: Feature Selection approach to detect Android malware using Deep Learning, 16–19. https://doi.org/10.1109/icsess47205.2019.9040821.

Mahindru, A., & Singh, P. (2017). Dynamic Permissions based Android Malware Detection using Machine Learning Techniques. Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC '17, (March 2018), 202–210. https://doi.org/10.1145/3021460.3021485.

Maiorca, D., Ariu, D., Corona, I., Aresu, M., & Giacinto, G. (2015). Stealth attacks: An extended insight into the obfuscation effects on Android malware. Computers and Security, 51, 16–31. https://doi.org/10.1016/j.cose.2015.02.007.

Masabo, E., Kaawaase, K. S., Sansa-Otim, J., Ngubiri, J., & Hanyurwimfura, D. (2020). Improvement of Malware Classification Using Hybrid Feature

Engineering. SN Computer Science, 1(1), 1–14. https://doi.org/10.1007/s42979-019-0017-9

Mathappan, N., & Rs, S. (2018). Comparative Analysis of Feature Selection Methods and Machine Learning Algorithms in Permission based Android Malware Detection, (December).

Mazlan, N. H., & Hamid, I. R. A. (2018). Evaluation of Feature Selection Algorithm for Android Malware Detection. International Journal of Engineering & Technology, 7(4.31), 311–315.

Milosevic, N., Dehghantanha, A., & Choo, K.-K. R. (2017). Machine learning aided Android malware classification. Computers & Electrical Engineering, 0, 1–9. https://doi.org/10.1016/j.compeleceng.2017.02.013.

Moosmann, F., Triggs, B., & Jurie, F. (2007). Fast discriminative visual codebooks using Randomized Clustering Forests. Advances in Neural Information Processing Systems, 985–992. https://doi.org/10.7551/mitpress/7503.003.0128.

Nguyen-Vu, L., Ahn, J., & Jung, S. (2019). Android Fragmentation in Malware Detection. Computers and Security, 87, 101573. https://doi.org/10.1016/j.cose.2019.101573.

Nivaashini, M., Soundariya, R. S., Vidhya Shri, H., & Thangaraj, P. (2018). Comparative analysis of feature selection methods and machine learning algorithms in permission based android malware detection. Proceedings of IEEE International Conference on Intelligent Computing and Communication for Smart World, I2C2SW 2018, 72–77. https://doi.org/10.1109/I2C2SW45816.2018.8997527.

Pekta, A. (2017). Stąpor K. (2018). Evaluating and Comparing Classifiers: Review, Some Recommendations and Limitations. In: Kurzynski M., Wozniak M., Burduk R. (eds) Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017. CORES 2017. Adv, (May). https://doi.org/10.1007/978-3-319-59162-9.

Pektaş, A., & Acarman, T. (2018). Ensemble machine learning approach for android malware classification using hybrid features. Advances in Intelligent Systems and Computing, 578(September), 191–200. https://doi.org/10.1007/978-3-319-59162-9_20.

Pektaş, A., & Acarman, T. (2020). Learning to detect Android malware via opcode sequences. Neurocomputing, 396, 599–608. https://doi.org/10.1016/j.neucom.2018.09.102.

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2018). TESSERACT: Eliminating experimental bias in malware classification across space and time. arXiv.

Peynirci, G., Eminağaoğlu, M., & Karabulut, K. (2020). Feature Selection for Malware Detection on the Android Platform Based on Differences of IDF Values. Journal of Computer Science and Technology, 35(4), 946–962. https://doi.org/10.1007/s11390-020-9323-x.

Pliakos, K., & Vens, C. (2016). Feature Induction based on Extremely Randomized Tree Paths. Nfmcp, 1–11. Retrieved from http://www.di.uniba.it/~loglisci/NFmcp2016/NFmcp2016_paper_14.pdf.

Qiao, M., Sung, A. H., & Liu, Q. (2016). Merging permission and api features for android malware detection. Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016, 566–571. https://doi.org/10.1109/IIAI-AAI.2016.237.

Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S., Wang, Y., & Xiang, Y. (2019). A3CM: Automatic Capability Annotation for Android Malware. IEEE Access, 7, 147156–147168. https://doi.org/10.1109/access.2019.2946392.

Rai, S., Dhanesha, R., Nahata, S., & Menezes, B. (2017). Malicious application detection on android smartphones with enhanced static-dynamic analysis. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10717 LNCS, 194–208. https://doi.org/10.1007/978-3-319-72598-7_12.

Rathore, H., Sahay, S. K., Rajvanshi, R., & Sewak, M. (2021). Identification of Significant Permissions for Efficient Android Malware Detection. Springer International Publishing. https://doi.org/10.1007/978-3-030-68737-3_3.

Ripon, S. H., Kamal, S., Hossain, S., & Dey, N. (2016). Theoretical Analysis of Different Classifiers under Reduction Rough Data Set. International Journal of Rough Sets and Data Analysis, 3(3), 1–20. https://doi.org/10.4018/ijrsda.2016070101.

Rs, S., Soundariya, N. M., & Thangaraj, R. S. (2018). Comparative Analysis of Feature Selection Methods and Machine Learning Algorithms in Permission based Android Malware Detection. Retrieved from https://www.researchgate.net/publication/329936217.

Sahal, A. A., Alam, S., & Sogukpinar, I. (2018). Mining and Detection of Android Malware Based on Permissions. UBMK 2018 - 3rd International Conference on Computer Science and Engineering, 264–268. https://doi.org/10.1109/UBMK.2018.8566510.

Sahin, D. O., Kural, O. E., Akleylek, S., & Kilic, E. (2018). New results on permission based static analysis for Android malware. 2018 6th International Symposium on Digital Forensic and Security (ISDFS),1–4. https://doi.org/10.1109/ISDFS.2018.8355377.

Şahin, D. Ö., Kural, O. E., Akleylek, S., & Kılıç, E. (2021). A novel Android malware detection system: adaption of filter-based feature selection methods. Journal of Ambient Intelligence and Humanized Computing, (123456789). https://doi.org/10.1007/s12652-021-03376-6.

Salah, A., Shalabi, E., & Khedr, W. (2020). A lightweight android malware classifier using novel feature selection methods. Symmetry, 12(5), 1–16. https://doi.org/10.3390/SYM12050858.

Salehinejad, S., Niakan Kalhori, S. R., Hajesmaeel Gohari, S., Bahaadinbeigy, K., & Fatehi, F. (2021). A review and content analysis of national apps for COVID-19 management using Mobile Application Rating Scale (MARS). Informatics for Health and Social Care, 46(1), 42–55. https://doi.org/10.1080/17538157.2020.1837838

Sangal, A. L. (2021). FSDroid : - A feature selection technique to detect malware from Android using Machine Learning. MULTIMEDIA TOOLS AND APPLICATIONS.

Saudi, M. M., Zahari, L. H., Ridzuan, F., Basir, N., Ali Pitchay, S., & Nabila, N. F. (2017). A New Mobile Malware Classification for Camera Exploitation based on System Call and Permission. World Congress on Engineering and Computer Science, I. Retrieved from http://www.iaeng.org/publication/WCECS2017/WCECS2017_pp95-100.pdf

Scherf, M., & Brauer, W. (1997). Feature Selection by Means of a Feature Weighting Approach (Technical Report No. FKI22197). Environment. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.8061&rep=rep1&type=pdf.

Sethi, K., Kumar, R., Sethi, L., Bera, P., & Patra, P. K. (2019). A novel machine learning based malware detection and classification framework. 2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019, (January 2020), 1–4. https://doi.org/10.1109/CyberSecPODS.2019.8885196.

Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. (2012). "Andromaly": A behavioral malware detection framework for android devices. Journal of Intelligent Information Systems, 38(1), 161–190. https://doi.org/10.1007/s10844-010-0148-x.

Shamim Ripon, Sarwar Kamal, Saddam Hossain, N. D. (2016). Theoretical Analysis of Different Classifiers under Reduction Rough Data Set : International Journal of Rough Sets and Data Analysis, 3(3), 1–20. https://doi.org/10.4018/IJRSDA.2016070101.

Shang, F., Li, Y., Deng, X., & He, D. (2017). Android malware detection method based on naive bayes and permission correlation algorithm. Cluster Computing, 21(1), 955–966. https://doi.org/10.1007/s10586-017-0981-6.

Sharma, K., & Gupta, B. B. (2018). Mitigation and risk factor analysis of android applications. Computers and Electrical Engineering, 71(March), 416–430. https://doi.org/10.1016/j.compeleceng.2018.08.003.

Sheen, S., Anitha, R., & Natarajan, V. (2015). Android based malware detection using a multifeature collaborative decision fusion approach. Neurocomputing, 151(P2), 905–912.https://doi.org/10.1016/j.neucom.2014.10.004.

Shhadat, I., Bataineh, B., Hayajneh, A., & Al-Sharif, Z. A. (2020). The Use of Machine Learning Techniques to Advance the Detection and Classification of Unknown Malware. Procedia Computer Science, 170(2019), 917–922. https://doi.org/10.1016/j.procs.2020.03.110.

Shrivastava, G., Kumar, P., Gupta, D., & Rodrigues, J. J. P. C. (2019). Privacy issues of android application permissions: A literature review. Transactions on

Emerging Telecommunications Technologies, (September), 1–17. https://doi.org/10.1002/ett.3773.

Singh, A. K., Jaidhar, C. D., & Kumara, M. A. A. (2019). Experimental analysis of Android malware detection based on combinations of permissions and API-calls. Journal of Computer Virology and Hacking Techniques, 15(3), 209–218. https://doi.org/10.1007/s11416-019-00332-z.

Singh, L., & Hofmann, M. (2018). Dynamic behavior analysis of android applications for malware detection. ICCT 2017 - International Conference on Intelligent Communication and Computational Techniques, 2018–Janua(2013), 1–7. https://doi.org/10.1109/INTELCCT.2017.8324010.

Sokolova, K., Perez, C., & Lemercier, M. (2017). Android application classification and anomaly detection with graph-based permission patterns. Decision Support Systems, 93, 62–76. https://doi.org/10.1016/j.dss.2016.09.006.

Su, D., Liu, J., Wang, W., Wang, X., Du, X., & Guizani, M. (2018). Discovering Communities of Malapps on Android-based Mobile Cyber-physical Systems. AD HOC NETWORKS.

Suarez-tangil, G., Dash, S. K., Ahmadi, M., Kinder, J., Giacinto, G., & Cavallaro, L. (2017). DroidSieve : Fast and Accurate Classification of Obfuscated Android Malware.

Surendran, R., Thomas, T., & Emmanuel, S. (2020). A TAN based hybrid model for android malware detection. Journal of Information Security and Applications, 54. https://doi.org/10.1016/j.jisa.2020.102483.

Taha, A. A., & Malebary, S. J. (2020). Hybrid classification of Android malware based on fuzzy clustering and the gradient boosting machine. Neural Computing and Applications. https://doi.org/10.1007/s00521-020-05450-0.

Taha, A. M., Mustapha, A., & Chen, S. Der. (2013). Naive Bayes-guided bat algorithm for feature selection. The Scientific World Journal, 2013(June 2014). https://doi.org/10.1155/2013/325973.

Taheri, R., Ghahramani, M., Javidan, R., Shojafar, M., Pooranian, Z., & Conti, M. (2020). Similarity-based Android malware detection using Hamming distance of static binary features. Future Generation Computer Systems, 105, 230–247. https://doi.org/10.1016/j.future.2019.11.034.

Talha, K. A., Alper, D. I., & Aydin, C. (2015). APK Auditor: Permission-based Android malware detection system. Digital Investigation, 13, 1–14. https://doi.org/10.1016/j.diin.2015.01.001.

Tangirala, S. (2020). Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. International Journal of Advanced Computer Science and Applications, 11(2), 612–619. https://doi.org/10.14569/ijacsa.2020.0110277.

Tuan, L. H., Cam, N. T., & Pham, V. H. (2019). Enhancing the accuracy of static analysis for detecting sensitive data leakage in Android by using dynamic analysis. Cluster Computing, 22(s1), 1079–1085. https://doi.org/10.1007/s10586-017-1364-8.

Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123–147. https://doi.org/10.1016/j.cose.2018.11.001.

Unal, Y., Polat, K., & Erdinc Kocer, H. (2014). Pairwise FCM based feature weighting for improved classification of vertebral column disorders. Computers in Biology and Medicine, 46(1), 61–70. https://doi.org/10.1016/j.compbiomed.2013.12.004.

Varsha, M. V., Vinod, P., & Dhanya, K. A. (2016). Identification of malicious android app using manifest and opcode features. Journal of Computer Virology and Hacking Techniques. https://doi.org/10.1007/s11416-016-0277-z.

Verma, sushma muttoo. (2016). An android malware detection framework based on permissions and intents. Defense Science Journal, 66(6), 618–623.

Vinod, P., Zemmari, A., & Conti, M. (2019). A machine learning based approach to detect malicious android apps using discriminant system calls. Future Generation Computer Systems, 94, 333–350. https://doi.org/10.1016/j.future.2018.11.021.

Wang, W., Li, Y., Wang, X., Liu, J., & Zhang, X. (2016). Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. Future Generation Computer Systems. https://doi.org/10.1016/j.future.2017.01.019.

Wang, W., Li, Y., Wang, X., Liu, J., & Zhang, X. (2018). Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. Future

Generation Computer Systems, 78, 987–994. https://doi.org/10.1016/j.future.2017.01.019.

Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., & Zhang, X. (2014). Exploring permission-induced risk in android applications for malicious application

detection. IEEE Transactions on Information Forensics and Security, 9(11), 1869–1882. https://doi.org/10.1109/TIFS.2014.2353996.

Wang, Z., Li, K., Hu, Y., Fukuda, A., & Kong, W. (2019). Multilevel permission extraction in android applications for malware detection. CITS 2019 - Proceeding of the 2019 International Conference on Computer, Information and Telecommunication Systems, 1–5. https://doi.org/10.1109/CITS.2019.8862060.

Wosiak, A., & Dziomdziora, A. (2015). Feature selection and classification pairwise combinations for high-dimensional tumour biomedical datasets. Schedae Informaticae, 24, 53–62. https://doi.org/10.4467/20838476SI.15.005.3027.

Xiao, X., Wang, Z., Li, Q., Xia, S., & Jiang, Y. (2017). Back-propagation neural network on Markov chains from system call sequences: A new approach for detecting Android malware with system call sequences. IET Information Security, 11(1), 8–15. https://doi.org/10.1049/iet-ifs.2015.0211.

Xu, Y., Wu, C., Zheng, K., Wang, X., Niu, X., & Lu, T. (2017). Computing Adaptive Feature Weights with PSO to Improve Android Malware Detection. Security and Communication Networks, 2017. https://doi.org/10.1155/2017/3284080.

Yan, P., & Yan, Z. (2018). A survey on dynamic mobile malware detection. Software Quality Journal, 26(3), 891–919. https://doi.org/10.1007/s11219-017-9368-4.

Yeom, C., & Won, Y. (2019). Vulnerability Evaluation Method through Correlation Analysis of Android Applications. Sustainability, 11(23), 6637. https://doi.org/10.3390/su11236637.

Yerima, S. Y., & Sezer, S. (2019). DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection. IEEE Transactions on Cybernetics, 49(2), 453–466. https://doi.org/10.1109/TCYB.2017.2777960Y.

Yerima, S. Y., Sezer, S., & Muttik, I. (2015). High accuracy android malware detection using ensemble learning. IET Information Security, 9(6), 313–320. https://doi.org/10.1049/iet-ifs.2014.0099.

Yuan, H., Tang, Y., Sun, W., & Liu, L. (2020). A detection method for android application security based on TF-IDF and machine learning. PLoS ONE, 15(9 September), 1–19. https://doi.org/10.1371/journal.pone.0238694.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. Journal of Applied Science and Technology Trends, 1(2), 56–70. https://doi.org/10.38094/jastt1224.

Zhang, Y., Ren, W., Zhu, T., & Ren, Y. (2019). SaaS: A situational awareness and analysis system for massive android malware detection. Future Generation Computer Systems, 95, 548–559. https://doi.org/10.1016/j.future.2018.12.028.

Zhou, Y., & Jiang, X. (2012). Dissecting Android malware: Characterization and evolution. Proceedings - IEEE Symposium on Security and Privacy, (4), 95–109. https://doi.org/10.1109/SP.2012.16.

Zhu, H. J., Jiang, T. H., Ma, B., You, Z. H., Shi, W. L., & Cheng, L. (2018). HEMD: a highly efficient random forest-based malware detection framework for Android. Neural Computing and Applications, 30(11), 3353–3361. https://doi.org/10.1007/s00521-017-2914-y.

Zhu, H. J., You, Z. H., Zhu, Z. X., Shi, W. L., Chen, X., & Cheng, L. (2018). DroidDet: Effective and robust detection of android malware using static analysis along with rotation forest model.Neurocomputing, 272, 638–646. https://doi.org/10.1016/j.neucom.2017.07.030.

LIST OF PUBLICATIONS

Indexed Journal with Impact Factors

1. **Abubaker, H**., Shamsuddin, S. M., & Ali, A. (2018). Analytics on malicious android applications. *International Journal of Advances in Soft Computing and Its Applications*, *10*(1), 106–118. **(Q4, IF: 0.794)**
2. **Alkaaf, H. A**., Ali, A., Shamsuddin, S. M., & Hassan, S. (2020). Exploring permissions in android applications using ensemble-based extra tree feature selection. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1), 543. https://doi.org/10.11591/ijeecs.v19.i1.pp543-552. **(Q3, IF: 1.294)**
3. **Abubaker, H**., Ali, A., & Shamsuddin, S. M. (2021). Weighted ensemble based extra tree for permission analysis for android applications, *99*(7), 1526–1536. http://www.jatit.org/volumes/Vol99No7/5Vol99No7. **(Q4, IF: 0.595)**

Indexed Conference Proceedings

1. **Al-Kaaf, H**., Ali, A., Shamsuddin, S., & Hassan, S. (2020). Feature selection for malicious android applications using Symmetrical Uncert Attribute Eval method. *IOP Conference Series: Materials Science and Engineering*, *884*(1). https://doi.org/10.1088/1757-899X/884/1/012060**.(Indexed by Scopus)**