This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

Detecting Mobile Malware Associated with Global Pandemics

*Alfredo J. Perez, University of Nebraska at Omaha, Omaha, NE, USA**

Sherali Zeadally, University of Kentucky, Lexington, KY, USA

David Tan, Valdosta State University, Valdosta, GA, USA

* Corresponding autor e-mail:alfredoperez@unomaha.edu

Abstract— More than 6 billion smartphones available worldwide can enable governments and public health organizations to develop apps to manage global pandemics. However, hackers can take advantage of this opportunity to target the public in nefarious ways through malware disguised as pandemics-related apps. A recent analysis conducted during the COVID-19 pandemic showed that several variants of COVID-19-related malware was installed by the public from non-trusted sources. We propose the use of app permissions and an extra feature (the total number of permissions) to develop a static detector using Machine Learning (ML) models to enable the fast-detection of pandemics-related Android malware at installation time. Using a dataset of more than 2000 COVID-19 related apps and by evaluating ML models created using decision trees and Naive Bayes, our results show that pandemics-related malware apps can be detected with an accuracy above 90% using decision tree models with app permissions and the proposed feature.

Introduction

The advent of global, real-time telecommunications along with the growth of mobile cellular technology in the last 25 years have helped to develop new alternatives to prepare for emergent diseases and their epidemics (and possibly pandemics). Various types of wearable/portable sensors that can be connected via Bluetooth to a smartphone provide an alternative to inform, diagnose, track, treat and manage epidemics and global pandemics [1]. With the emergence of the COVID-19 pandemic, more than 2,000 COVID-themed mobile apps (not including malware) were developed for different purposes around the world as of December of 2020 [2].

The COVID-19 pandemic has been exploited by cybercriminals using different threats, attacks and channels including Distributed Denial of Services Attacks (DDoS), malicious domains and websites, malware, ransomware, spam emails, malicious social media messaging, business email compromise, mobile apps, and browsing apps [3], impacting healthcare systems, financial services, government and media outlets, and the public. Cybercrime increased dramatically during the COVID-19 pandemic, with an estimated impact of more than 6 trillion US dollars worldwide in 2021 [4]. This major increase in cybercrime activity during 2021 was due to the massive online activity caused by worldwide lockdowns and restrictions in movement to mitigate the COVID-19 pandemic disease [4], and was performed by not only solitaire hackers and hacking groups, but also by major state-sponsored cybercriminals.

The availability of more than 6 billion smartphones during the COVID-19 pandemic [1] and their use in future pandemics (and other public health emergencies) make them an attractive target for hackers to release malware disguised as pandemics-related apps through non-trusted channels

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

(e.g., via social media, SMS/MMS, websites) fueled by disinformation. Moreover, pandemic-related malware installed during emergencies could be used to enable DDoS attacks on other systems. Thus, to prepare for future pandemics we must develop systems to help mitigate the effects of emergent diseases and protect the global cyberinfrastructure during the containment and mitigation of pandemics.

In this study, we seek to answer the following research question: can pandemic-related malware be detected using a static detector based on Android permissions and machine learning methods.

Research contributions of this work

We summarize our research contributions as follows:

- We review mobile and smartphone use cases and how cybercriminals can exploit mobile apps during epidemics and pandemics.
- We propose and evaluate the use of Android app permissions combined with Machine Learning (ML) to enable static detectors for the fast detection of pandemics-related mobile malware at the edge.
- We propose the use of the total number of app permissions as an extra feature in the permission-based static detector (in addition to the app permissions themselves) as an approach to increase the accuracy of the detection.

We organize the rest of the paper as follows. In the next section we review related works. Later we present use cases of mobile phones and smartphones apps in epidemics/pandemics. Then we review mobile malware during pandemics with a focus on the COVID-19 pandemic. Next, we propose the use of app permissions and Machine Learning (ML) as a fast approach to detect malware in Android smartphones. Finally, we make some concluding remarks and some final recommendations on protection to smartphone users in future pandemics.

Related Work

Although the use of general Android permissions to detect malware has been proposed in the past using permissions with machine learning [5], permissions with Application Programming Interface (API) calls/graphs [6], comparison of permission patterns [7], intents and permissions [8], using multiple detectors and observation windows [9], and other techniques combining permissions with other static and dynamic approaches [10], these works were developed before the onset of COVID-19 pandemic when there was no knowledge on how global pandemic themed apps (both benign and malign) were implemented.

During the COVID-19 pandemic, Wang et al. researched and collected a dataset of COVID-19-themed apps and types (e.g., malware and not malware) with more than 2000 unique apps collected from trusted and non-trusted sources including both benign and malign apps [2] and analyzed their permissions. In their work, Wang et al. did not propose specific approaches to detect malware for pandemic-related apps. Similarly, Sun et al. [11] analyzed the security and privacy of 34 COVID-19 contact tracing apps with the goal of recommending security practices in the development of COVID-19-themed apps, and the development of a tool called COVIDGUARDIAN based on

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

static analysis and data flow analysis to find vulnerabilities of trusted COVID-19-themed apps. Recently Manzil and Naik [12] used app permissions with machine learning and achieved an accuracy of 81% and 83% respectively with a dataset of 100 app samples using random forest and decision trees. It is worth pointing out that the work of Manzil and Naik focused only on COVID-19-themed apps, and their work is the most similar one to ours but with the following differences: (1) we used a bigger dataset of COVID-19 related malware to train our models; (2) we proposed and explain why the use of the total number of permissions as an extra feature (in addition to the app permissions) is helpful in detecting pandemics-related malware; and (3) our approach, while targeting specific type of apps (pandemics-related malware), does not incur significant overhead for detection when compared with other approaches (e.g., the approach proposed by Ficco [9] that requires more features and an ensemble of ML models to detect malware, or the use of permissions with Intents and API calls [13]) that are more resource intensive for the Android Operating System's (OS) .

Pandemics and Cellphones/Smartphones and their Limitations

Smartphone Apps' Use Cases During Pandemics/Epidemics

Recently, the COVID-19 pandemic has highlighted the use smartphones as tools which can be used to manage public health emergencies. However, past epidemics and pandemics had leveraged the use of smartphones and data generated by mobile cellular communications (Table 1). For example, in 2003 a Hong Kong mobile operator launched a Location-Based Service (LBS) via Short Messaging Service (SMS) and Wireless Application Protocol (WAP) to notify subscribers when a nearby building was contaminated with the Severe Acute Respiratory Syndrome (SARS) during the 2003 SARS outbreak in Asia [14]. Radio Frequency IDentification (RFID) was used during this SARS outbreak in Singapore for contact tracing inside hospitals [14], allowing health officials to identify 10 times faster who an infected person had contact with than using other methods. A similar approach was used during COVID-19 in different parts of the world using Bluetooth Low Energy (BLE) [6].

Table 1. Mobile and smartphone's use cases during epidemics and pandemics in healthcare-related applications

Use case	Epidemic/pandemic disease	Approach example
Location-Based Service (LBS) for building infection notification	Severe Acute Respiratory Syndrome (SARS) outbreak	Use of SMS and WAP to notify subscribers during the 2003 SARS outbreak in Hong Kong [14]
Contact tracing	SARS and COVID-19	Use of RFID to conduct contact tracing in Singaporean hospitals during the 2003 SARS outbreak [14]

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

		Use of Bluetooth Low-Energy during the COVID-19 pandemic in contact-tracing apps worldwide [15]
Surveillance and tracking of virus spread	Cholerae	Surveillance of cholerae in wide areas using anonymized mobile cellular operators' data from the 2010 Haiti cholerae outbreak [16]
Disease detection	Zika, Chikungunya, and Dengue Malaria, Ebola, and Marburg virus disease	Use of portable devices and sensors used with smartphones to detect pathogens in human specimens [17][18][19]
Treatment adherence and long-term disease management	Human Immunodeficiency Virus (HIV) and tuberculosis	Use of Short Message Service (SMS) text messages in Kenya to remember patients to take AntiRetroviral Therapy (ART) medication [20] Similar approaches in other African nations for both HIV and tuberculosis [21]
Health education	HIV, tuberculosis, COVID-19	Mobile applications used by public health organizations to inform the public about infectious diseases, their symptoms, and effects [21]
Digital Health Passports (DHPs)	COVID-19	Mobile applications used by airlines, the European Union (EU), USA, private organizations, and other countries (e.g., Israel) to grant access to services for COVID-19 vaccinated individuals [22]
Telemedicine and communication between patients and families	COVID-19	Used extensively in the world during the COVID-19 pandemic for patients to contact practitioners due to lockdowns and safety precautions [1] Used by hospitalized patients in the world to contact their families to minimize contamination risks.

Using only anonymized mobile phone data from cellular operators, Bengtsson et al. [16] created a model to survey and track the spread of cholerae in the 2010 Haiti epidemic. Their research showed

that mobile operators' data can help to track and contain the spread of infectious diseases and serve as a surveillance mechanism for wide areas. In 2017, Priye et al. [17] reported on the rapid detection of Zika, Chikungunya, and Dengue viruses using a portable device called the "LAMP Box", a smartphone's camera, and an app to detect and analyze samples of human specimens (e.g., blood, urine, and saliva). In a similar way, Yu et al. developed a smartphone app to detect the presence of Malaria parasites *(P. falciparum)* on digital photos captured using a smartphone's camera placed on a microscope's eyepiece lens when a user places a slide with human blood specimens to be examined under the microscope [18]. Natesan et al. [19] developed a similar approach for the detection of Ebola and Marburg viruses.

To adhere to treatment using AntiRetroviral Therapy (ART) for Human Immunodeficiency Virus (HIV) management, in 2012, Horvath et al. [20] studied the use of mobile phone SMS text messaging and they found based on two Randomized Controlled Trials (RCTs) studies in Kenya that weekly mobile phone text-messaging improved HIV viral load suppression by reminding patients to take their medications, thus helping them to adhere to their therapy. For long-term treatment, Devi et al. [21] found in a literature (covering the period 2005 to 2015) review on long-term care/management of HIV/AIDS and tuberculosis that mobile phones were successfully used for long-term care and management of these diseases in developing countries. They reported that 73.3% of their reviewed papers (66 papers) reported positive effects on HIV/tuberculosis management using mobile phones. Finally, during the COVID-19 pandemic other use cases of smartphones (and tablets) apps for public health settings included telemedicine/patient communication, health education, and apps implementing Digital Health Passports (DHPs) [22].

Limitations of Smartphones and their applications during Pandemics

While there have been great advances on the use of smartphones for epidemics/pandemics, there are also limitations for the successful use of smartphone apps during epidemics/pandemics in aspects such as interoperability, effectiveness, politics, design choices and marketing, and security and privacy.

From the interoperability perspective, applications developed during pandemics with a healthcare (or fitness) focus use a particular architecture (in hardware or software) that forbids (or makes it almost impossible) for users to switch components (e.g., wearables for monitoring), health providers, or move healthcare data collected through them. While limitations may be related to laws, others are related to the lack of standardization and business models that makes it difficult to achieve interoperability among systems [1].

Many apps developed during pandemics are not evaluated for their effectiveness before or after deployment. For example, Devi et al. [21], in their research about long-term treatment with mobile apps for HIV and tuberculosis, found that many research studies lack statistical evaluations on app effectiveness and rather used casual/anecdotal observations. The lack of evaluation is exacerbated by the need for rapid development of many mobile apps that are created as a public health response aimed at an emergent disease (e.g., COVID-19 case), thus impacting an app's efficacy, reliability, and privacy/security.

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

Additionally, the implementation of certain types of smartphone apps for pandemics may be subjected to politics. For example, during the COVID-19 pandemic, vaccination passports and their smartphone implementations (through DHPs) were subjected to policy decisions that varied between U.S. states. DHPs were implemented in the state of New York when COVID-19 vaccination became widely available [22]. However, in Florida any kind of vaccination passport was forbidden by an executive order from Governor Ron DeSantis in April 2021 [23].

From the perspective of the design and marketing of apps, the approach used to develop, implement, promote, and give choices to the public about pandemics-related apps may affect their installation. In this context, a recent survey in the U.S. with 1963 respondents which studied why somebody would install a contact tracing app for COVID-19 (by exploring the design space of contact tracing apps), Li et al. [24] found that the developers' choice on app design and users' individual differences (e.g., users' job/work, income, demographics, use of public transportation, technology readiness) have a significant impact on whether a person will install a contact tracing app over other factors such as app's security and privacy. They recommend highlighting the public health benefit as a leverage to promote contact tracing apps and paying attention to apps' design and marketing strategies among essential/health workers because their higher vulnerability to contract an emergent infectious disease such as COVID-19, and people living in rural areas because their lower preference on installing contact tracing apps developed by large private companies.

Finally, short software development cycles used to develop and launch applications/systems during pandemics can result in data leak (affecting the privacy and security of users) and make software systems developed during pandemics vulnerable to cyberattacks. Results from a study done during the COVID-19 pandemic in 2021 showed that 78% of the companies surveyed believed their technical debt increased during 2021, with most of the technical debt believed to be arising from the development of new products [25]. In the same survey, 86% of respondents mentioned that launching new digital products/services justified the technical debt incurred.

Mobile Malware during Pandemics

Mobile apps developed before COVID-19 for epidemics/pandemics were mostly applications developed by well-known organizations as part of health campaigns or prototype systems. However, the worldwide availability of smartphones and other wearables at the start of the COVID-19 pandemic, and their increasing use as the pandemic progressed [1], made smartphones an attractive target of malware which grew quickly during the COVID-19 pandemic.

More than 2 million installations of mobile malware packages were performed worldwide during the fourth quarter of 2020, which almost doubled the number of malware package installations during the third quarter of the same year (around 1.1 million in the third quarter of 2021) [26]. These numbers began to decrease during 2021, reaching around nine hundred thousand installations by the second quarter of 2021 (Figure 1). Hackers also exploited users through malware camouflaged as legitimate COVID-19-themed apps. There were at least 370 unique COVID-19-themed mobile malware apps developed worldwide as of mid-November 2020 targeting the Android operating system with most apps released after March 2020 [2].

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

Hackers targeted smartphones during the COVID-19 pandemic not only because of their ubiquitous use, but also because of the lack of cybersecurity hygiene of smartphone users around the world. Misinformation and mobile malware distribution methods (different from the use of app stores), and vulnerabilities such as SMS phishing (by which SMS messages are used to distribute malware) and Zero-Click attacks (by which no input from users is needed before deploying an attack, but rather by exploiting vulnerabilities in apps already installed) were used by hackers to launch attacks on smartphone users during COVID-19 [27].

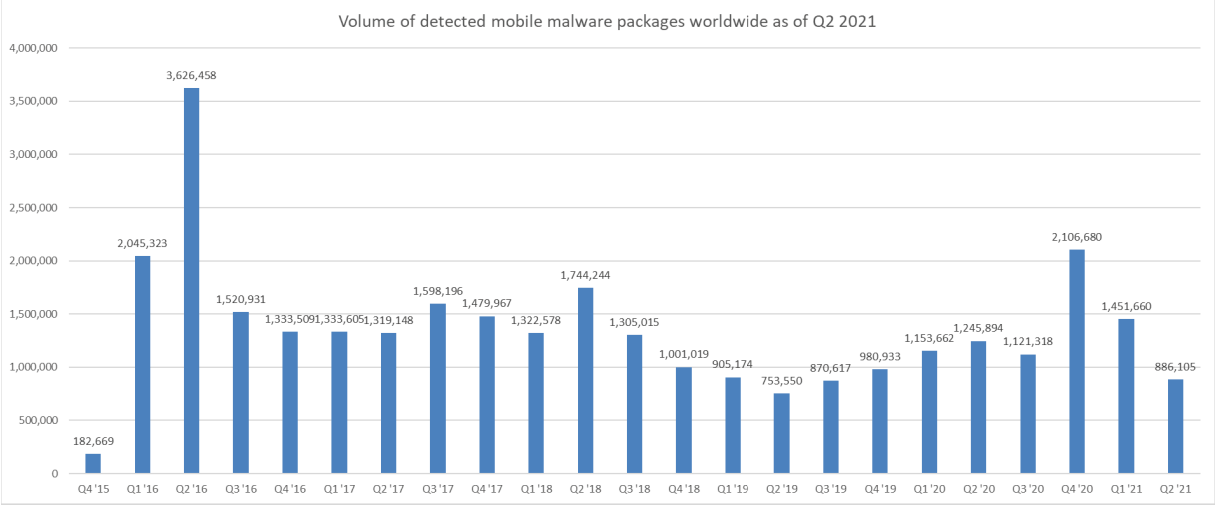

Figure 1. Mobile malicious installation packages detected from Q4 2015 to Q2 2022 based on data from Karpersky Lab's Securelist [27]

Other distribution mechanisms for mobile malware during COVID-19 included messages sent via social media apps (e.g., WhatsApp, Instagram, and others), and camouflaged malware distributed via app stores for both Android and iOS devices (i.e., Google Play Store and Apple App Store), even though app stores blocked more than 1 million attempts to circumvent security measures to publish mobile apps [27].

According to Karpsersky data, most of the new worldwide mobile malware in 2021 was in the form of AdWare (42.42% of the total), RiskTool (by which malware conceals files, run apps silently, or terminate active process, 35.27% of the total), and trojans (programs that claim to perform some function while doing something else, 8.86% of the total) [28]. For COVID-19-themed malware, Wang et al. [2] reported that trojans (56%) and spyware (29%) made most of the COVID-19-themed malware in Android as of November 2020. Ransomware made about 7% of mobile malware in their study.

Detecting Pandemics-related Malware Using App Permissions and Machine Learning

In this section we describe the use of Android app permissions and Machine Learning (ML) to detect COVID-19-themed malware. We describe our dataset, the ML models trained, and we evaluate the models' performance, and discuss our results.

Permissions Dataset

We obtained our dataset by extracting Android app permissions from the COVID-19-themed apps collection curated by Wang et al. [2]. Their collection (made publicly available by its curators) has 2,500 unique Android Package Kits (APKs) with 370 unique APKs belonging to malicious apps collected by mid-November 2020. We used AndroGuard to extract each apps' permissions from their corresponding APK's manifest.

Due to errors generated by AndroGuard when extracting app permission data from some of the APKs, we ended with permissions of 2016 unique apps (80% of the original dataset) with 277 labeled as COVID-19-themed malware samples (75% of the original malware samples) and 1739 labeled as non-malware COVID-19-themed apps (81% of the original non-malware samples). We extracted 203 unique permissions from all apps in our dataset. We created a spreadsheet with each row storing the permissions used by an app. Each column had a "1" or "0" depending on the use of a permission by an app. We also used two more columns specifying the apps' name and its type/class (malware or not). For example, if an app is a malware app and uses all permissions, then in the row for that app we store in all columns a value of "1". Otherwise, if a second app did not use any permission and it is not malware, then in the row for that second app we store a value of "0" in all columns. Figure 2 illustrates how we stored the permissions in the spreadsheet for the *Stato COVID-19 Italia* Android app. This app uses four permissions, thus a value of "1" appears in each of the corresponding permission columns. As this app is not malware, a "0" appears in the MALWARE column in figure 2. The resulting spreadsheet has 2016 rows (one row per app) and 205 columns (203 columns for permissions plus two more for the app name and the class/type). This spreadsheet is the dataset we used to train the models. When using the models, the static detector extracts the permissions used by the app in the manifest at installation time, counts the number of permissions used by the app, and then executes a trained ML model. This process does not cause additional overhead for the OS because the Android security model extracts the permissions from the manifest at installation time.

We created the chart shown in figure 3 using our dataset. This chart shows that around 30% of non-malware apps used one permission, while most of the malware apps used four or more app permissions. The sample mean of the number permissions used by malware apps was $u_0 = 7.09$ ± 3.6 permissions per app, and the sample mean of the number of permissions used by non-malware apps was $u_1 = 1.8$ ± 1.5 permissions per app.

Assuming a normal distribution on the total number of permissions per app for both malware and non-malware apps, we conducted a t-test with H_0: $u_0 - u_1 \leq 0$ (null hypothesis: both app classes use the same number of permissions), H_1: $u_0 - u_1 > 0$ (alternative hypothesis: malware apps use more permissions than non-malware), $\alpha = 0.1$ (statistical significance), and the result was that H_0 (null hypothesis) was rejected (H_0 was even rejected at a statistical significance value $\alpha = 0.01$). This

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

result suggests that the total number of permissions used per app can be added as an extra feature to detect COVID-19-themed malware.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="org.twistedappdeveloper.statocovid19italia">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

App permissions in the manifest

APP_NAME	MALWARE	ACCEPT_HANDOVER	...	ACCESS_NETWORK STATE	...	FOREGROUND_SERVICE	...	INTERNET	...	RECEIVE_BOOT_COMPLETED	...
...	...	...	...	...	...	...	...	...	...	...	...
Stato COVID-19 Italia	0	0		1		1		1		1	
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...

Spreadsheet with permissions from all apps

Figure 2. Permissions used by a COVID-19-themed app and its row in our spreadsheet

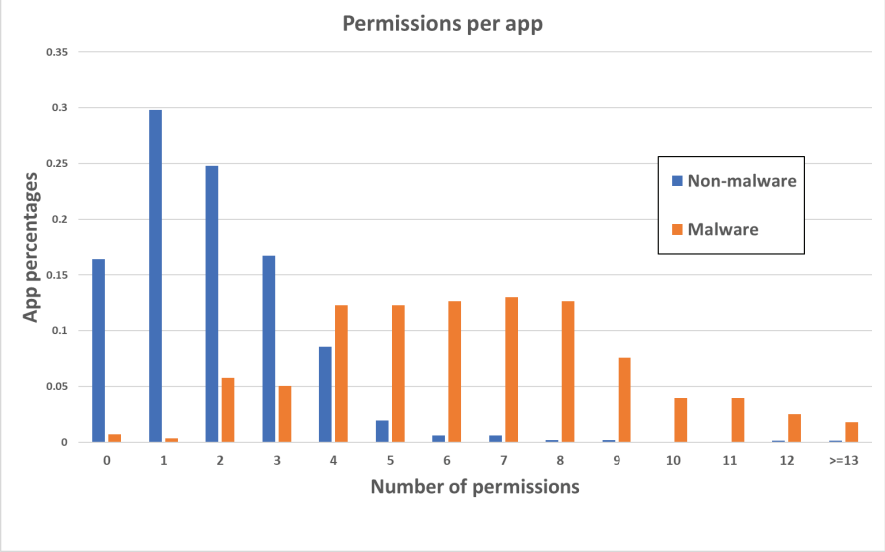

Figure 3. Distribution of total number of app permissions per app class (malware/non-malware) for COVID-19-themed Android apps.

Model Creation

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

The problem of malware detection can be modeled as a machine learning classification problem with two target classes (malware/non-malware). To create the input data for the models we added the total number of permissions used by an app as an extra feature (in addition to each app's permissions).

We used Weka 3.8.6 to train the machine learning classification models. We trained the first set of three models using the spreadsheet dataset obtained in the previous section (2016 instances, 203 app permissions, and the extra column/feature for the total number of permissions), and we trained a second set of three ML classification models using an augmented dataset with 3955 instances with the same number of app permissions/features (203 app permissions and the extra feature for total number of app permissions). We ignored the name of the app when training the models.

We created the second (augmented) dataset using the Synthetic Minority Oversampling TEchnique (SMOTE) [20] to balance the number of malware instances of the original dataset. SMOTE [29] is a machine learning technique to generate synthetic data when it is desired build classification models for two classes, and one class has significantly less samples compared to the other class (malware samples/instances vs non-malware samples/instances in our case). SMOTE generates synthetic instances for the minority class that are plausible (i.e., better than duplicating the instances in the minority class).

The SMOTE permissions dataset had 3955 instances (1739 for the non-malware class and 2216 for the malware class). While this second dataset had more instances for the malware class (1739 non-malware instances vs 2216 malware instances), the dataset is more balanced than the original spreadsheet dataset (1739 non-malware instances vs 277 malware instances).

We created three classification models with both datasets (six models in total) using OneRule, J48, and Naive Bayes algorithms. The OneRule and J48 algorithms build decision tree classification models, while the Naïve Bayes builds a probabilistic model for classification. OneRule creates a simple classification tree based only on the attribute/feature with the smallest total error as the selected attribute/feature to build the classification model. We selected these models because once the models are trained, they can be executed very quickly on a smartphone.

Performance evaluation of proposed models

We used a 10-fold cross validation on each algorithm with a 66% split for each fold (i.e., we used 66% of the instances on each fold for model training we used the other 34% to evaluate the performance of the models on each fold). We trained and evaluated the models using Weka 3.8.6 on a Windows-based Asus laptop equipped with an AMD Ryzen 7 processor running at 2.3 GHz and 16GB RAM. For each dataset, algorithm and class, we computed the following performance metrics:

- *True positive rate (TP Rate):* This is the probability that an instance will be correctly classified.
- *False positive rate (FP Rate):* This is the probability that an instance will be incorrectly classified. In our case this means that an app that is malware is classified as non-malware and vice versa.

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

- *Precision:* Proportion of actual instances correctly identified within each class.
- *F-Measure:* A measure of a model's accuracy. It is calculated from the precision and recall. Values close to 1 means better scores.

Table 2. Evaluation metric results for our classification models

Algorithm + Dataset	Class	TP Rate	FP Rate	Precision	F-Measure
OneRule + SMOTE	Non-malware	0.877	0.228	0.751	0.809
	Malware	0.772	0.123	0.889	0.826
J48 + SMOTE	Non-malware	0.98	0.031	0.961	0.971
	Malware	0.969	0.02	0.984	0.977
Naive Bayes + SMOTE	Non-malware	0.939	0.136	0.844	0.889
	Malware	0.864	0.061	0.948	0.904
OneRule	Non-malware	0.971	0.343	0.947	0.959
	Malware	0.657	0.029	0.781	0.714
J48	Non-malware	0.983	0.217	0.966	0.974
	Malware	0.783	0.017	0.879	0.828
Naive Bayes	Non-malware	0.965	0.119	0.981	0.973
	Malware	0.881	0.035	0.8	0.838

The performance evaluation presented in Table 2 shows the results of using a 10-fold cross validation which is an accepted methodology to evaluate ML models. In this table, we present the average values for each measure after training and evaluating each model 10 different times with random folds for training and testing the models.

Discussion of results

When creating our models using OneRule, we found that the *total number of permissions* was selected as the attribute/feature to build the OneRule models, meaning that this attribute alone is the best one to potentially detect a COVID-19-themed app as malware or not. We expected this result from our analysis of the sample means of the total number permissions for malware/non-malware apps (as figure 3 shows) and the statistical analysis we performed on the sample means. We observed that, in general, all algorithms performed relatively well but models based on the OneRule yielded the worst results when compared with J48 and Naïve Bayes models.

We obtained the best overall result when using SMOTE with the J48 decision tree algorithm. This model (identified as *J48+SMOTE* in table 2) had the best classification results for all the evaluation metrics, especially those associated with the malware class among all models. Our results show that a static malware detector specifically targeting pandemic-themed apps can be implemented directly in the Android OS because the OS detects the permissions during the APK installation. This detector can be used during pandemic times, specifically when users attempt to install apps from non-trusted sources, which was frequently the case of COVID-19-themed malware.

Although we did not implement our proposed methods on an actual smart phone to measure the time or the power consumption to execute a detection (we will conduct these measurements in the future), we argue that the computational and power/energy costs to detect pandemics-related malware based on permissions on a smart phone does not produce significant overhead because (1) Android extracts permissions from app manifests when an app is installed (the OS can run a malware detector based on permissions at installation time), and (2) the computational complexity of the ML models we tested (OneRule, J48 and Naive Bayes) execute in constant time (*O(1)*) because no extra work is done to extract the features as either a constant number of *if* statements is needed with basic Boolean expressions (to implement OneRule and J48), or a constant number of floating point multiplications is needed (Naive Bayes), which are approximately 508 multiplications (2 classes * 204 features) of single precision floating point operations at installation time.

Conclusion

We have reviewed the use of smartphones and their use cases during pandemics. We also reviewed pandemic-related malware trends during the COVID-19 pandemic. We evaluated the use of permissions and machine learning methods to detect COVID-19-themed malware and we found that a static malware detector can be developed in Android to detect pandemics-related malware with an accuracy of more than 90% using a combination of SMOTE, app permissions, total number of permissions, and decision trees. Moreover, from our review of COVID-19 related malware, we recommend the following countermeasures to minimize the impact of cyberattacks on smartphone users in future pandemics and global crises:

- Implement a static malware detector as part of the mobile OS as a software update during pandemics that can detect and alert about possible malware being installed from a non-traditional source (e.g., apps downloaded via SMS links or message links in social networks that camouflage malware as pandemic-related apps) or a non-trusted source.
- Increase the training and awareness of cybersecurity and cyberhygiene specifically focused on cybersecurity for smartphones during a pandemic. This could be achieved by cybersecurity education before a pandemic, and public announcement about mobile malware risks during a pandemic to diminish spear phishing attacks and avoid the installation of pandemics-related malware.
- Recommend that any kind of mobile app to be installed during a pandemic to be installed from a trusted source (e.g., Google Play Market, Apple App Store). This makes mobile malware to be harder to distribute and be installed, especially during pandemics and other global crises.

Acknowledgement

This work was supported by the U.S. National Science Foundation under grant awards 1950416 and 2308741. We thank the anonymous reviewers for their valuable comments, which helped improve the paper's content, quality, and organization.

References

[1] Perez, A. J., & Zeadally, S. (2021). Recent advances in wearable sensing technologies. Sensors, 21(20), 6828.

[2] Wang, L., He, R., Wang, H., Xia, P., Li, Y., Wu, L., ... & Xu, G. (2021). Beyond the virus: a first look at coronavirus-themed Android malware. Empirical Software Engineering, 26(4), 1-38.

[3] Khan, N. A., Brohi, S. N., & Zaman, N. (2020). Ten deadly cyber security threats amid COVID-19 pandemic.

[4] TechXplore. Global cost of cybercrime topped $ 6 trillion in 2021: defence firm (2022). Available online: https://techxplore.com/news/2022-05-global-cybercrime-topped-trillion-defence.html

[5] Zarni Aung, W. Z. (2013). Permission-based android malware detection. International Journal of Scientific & Technology Research, 2(3), 228-234.

[6] Peiravian, N., & Zhu, X. (2013, November). Machine learning for android malware detection using permission and api calls. In 2013 IEEE 25th international conference on tools with artificial intelligence (pp. 300-305). IEEE.

[7] Wang, C., Xu, Q., Lin, X., & Liu, S. (2019). Research on data mining of permissions mode for Android malware detection. Cluster Computing, 22, 13337-13350.

[8] Khariwal, K., Singh, J., & Arora, A. (2020, July). IPDroid: Android malware detection using intents and permissions. In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) (pp. 197-202). IEEE.

[9] Ficco, M. (2021). Malware analysis by combining multiple detectors and observation windows. IEEE Transactions on Computers, 71(6), 1276-1290.

[10] Pimenta, T. S. R., Ceschin, F., & Gregio, A. (2023). ANDROIDGYNY: Reviewing clustering techniques for Android malware family classification. Digital Threats: Research and Practice.

[11] Sun, R., Wang, W., Xue, M., Tyson, G., Camtepe, S., & Ranasinghe, D. C. (2021, May). An empirical assessment of global COVID-19 contact tracing applications. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE) (pp. 1085-1097). IEEE.

[12] Manzil, H. H. R., & Naik, M. S. (2022, January). COVID-Themed Android Malware Analysis and Detection Framework Based on Permissions. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-5). IEEE.

[13] 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.

[14] Eysenbach, G. (2003). SARS and population health technology. Journal of Medical Internet Research, 5(2), e882.

[15] Ahmed, N., Michelin, R. A., Xue, W., Ruj, S., Malaney, R., Kanhere, S. S., ... & Jha, S. K. (2020). A survey of COVID-19 contact tracing apps. IEEE access, 8, 134577-134601.

[16] Bengtsson, L., Gaudart, J., Lu, X., Moore, S., Wetter, E., Sallah, K., ... & Piarroux, R. (2015). Using mobile phone data to predict the spatial spread of cholera. Scientific reports, 5(1), 1-5.

[17] Priye, A., Bird, S. W., Light, Y. K., Ball, C. S., Negrete, O. A., & Meagher, R. J. (2017). A smartphone-based diagnostic platform for rapid detection of Zika, chikungunya, and dengue viruses. Scientific reports, 7(1), 1-11.

[18] Yu, H., Yang, F., Rajaraman, S., Ersoy, I., Moallem, G., Poostchi, M., ... & Jaeger, S. (2020). Malaria Screener: a smartphone application for automated malaria screening. BMC Infectious Diseases, 20(1), 1-8.

[19] Natesan, M., Wu, S. W., Chen, C. I., Jensen, S. M., Karlovac, N., Dyas, B. K., ... & Ulrich, R. G. (2018). A smartphone-based rapid telemonitoring system for Ebola and Marburg disease surveillance. ACS sensors, 4(1), 61-68.

This article has been accepted for publication in IEEE Pervasive Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MPRV.2023.3321218

[20] Horvath, T., Azman, H., Kennedy, G. E., & Rutherford, G. W. (2012). Mobile phone text messaging for promoting adherence to antiretroviral therapy in patients with HIV infection. Cochrane Database of Systematic Reviews, (3).

[21] Devi, B. R., Syed-Abdul, S., Kumar, A., Iqbal, U., Nguyen, P. A., Li, Y. C. J., & Jian, W. S. (2015). mHealth: An updated systematic review with a focus on HIV/AIDS and tuberculosis long term management using mobile phones. Computer methods and programs in Biomedicine, 122(2), 257-265.

[22] Gostin, L. O., Cohen, I. G., & Shaw, J. (2021). Digital health passes in the age of COVID-19: Are "vaccine passports" lawful and ethical?. JAMA, 325(19), 1933-1934.

[23] State of Florida, Office of the Governor. Executive Order Number 21-81 (Prohibiting COVID-19 Vaccine Passports) (2021). Available online: https://www.flgov.com/wp-content/uploads/2021/04/EO-21-81.pdf

[24] Li, T., Cobb, C., Yang, J. J., Baviskar, S., Agarwal, Y., Li, B., ... & Hong, J. I. (2021). What makes people install a COVID-19 contact-tracing app? Understanding the influence of app design and individual difference on contact-tracing app adoption intention. Pervasive and Mobile Computing, 75, 101439.

[25] Doerrfeld, B. A pandemic side effect: Rampant technical debt (2022). Available online: https://devops.com/a-pandemic-side-effect-rampant-technical-debt/

[26] Statistica, Number of detected malicious installation packages on mobile devices worldwide from 4th quarter 2015 to 2nd quarter 2021 (2021). Available online: https://www.statista.com/statistics/653680/volume-of-detected-mobile-malware-packages/

[27] Check Point Blog. The mobile malware landscape in 2022 – Of Spyware, Zero-Click attacks, Smishing and Store Security (2022). Available online: https://blog.checkpoint.com/2022/09/15/the-mobile-malware-landscape-in-2022-of-spyware-zero-click-attacks-smishing-and-store-security/

[28] Statistica, Distribution of new mobile malware worldwide in 2021, by type (2021). Available online: https://www.statista.com/statistics/653688/distribution-of-mobile-malware-type/

[29] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Biographies

Alfredo J. Perez is an Associate Professor with the University of Nebraska at Omaha (Omaha, NE). He received his B.Sc. in Systems Engineering from Universidad del Norte (Barranquilla, Colombia), and his M.Sc. and Ph.D. degrees from the University of South Florida (Tampa, FL). His research interests include mobile/ubiquitous computing and sensing, privacy and cybersecurity, and CS education. He is an IEEE Senior Member and a member of the National Academy of Inventors. Contact him at alfredoperez@unomaha.edu.

Sherali Zeadally received his doctoral degree in computer science from the University of Buckingham, England, followed by postdoctoral research at the University of Southern California, Los Angeles, CA. He is a Professor in the College of Communication and Information at the University of Kentucky. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England. Contact him at szeadally@uky.edu.

David Kingsley Tan is a M.S. student in the Department of Computer Science at the Georgia Institute of Technology (Atlanta, GA), and he is a software engineer with the Space Dynamics Laboratory. Tan received his B.Sc. degree in computer science from Valdosta State University (Valdosta, GA). Contact him at dtan68@gatech.edu.