アプリケーションの利用権限や特徴量からなる評価用データセットの構築についての考察

2017SE060 小椋愛実 2017SE103 山田麻佑衣
指導教員：横森励士

1 はじめに

近年スマートフォンが急速に普及し，スマートフォン上で動作させるアプリケーションの需要が増加している．大量のアプリケーションの中には悪意を持った昨日を内包しているものも多く存在する．それらによって，様々な脅威が日々引き起こされている．アプリケーションストアの管理者によって削除されることが多いが，削除までに利用してしまうと被害がもたらされる．ユーザの観点からは，アクセス権限など事前に公開される情報を利用して，そのようなアプリケーションを利用しないことが求められる．過去の研究では，アプリケーションの紹介ページで確認できる情報や，アプリケーションが要求する権限を取得しその情報に基づいて機械学習を行うことで，今後消される可能性が高いアプリケーションかどうかを判定する仕組みを提案した．情報を入手する仕組みまでは提案されているが，その仕組みに基づいて実際にデータを効率的に入手することはできなかった．

本研究では，過去の研究で提案された手法に基づいて実際にデータ収集を行う．ソフトウェアのリストアップからデータ収集までの期間を短くすることで手法が実現可能であることを確かめる．更に，データ収集の自動化について考察を行う．現在の仕組みでは，一部のデータのみ抽出の自動化が考慮されているが大部分のデータに関しては考慮されていない．Web スクレイピングの技術を用いて，どれくらいの情報が自動的に収集可能となりそうかを確認する．これらの実験や考察を通じて，実際に運用可能なシステムの構築を目指す．

2 背景技術

2.1 Android アプリケーションが配布される仕組みと権限について

代表的な Android アプリケーションの配布サービスとして，Google Play[1] が提供されている．アプリケーションを提供する側がアプリケーションとともにタイトル名やアプリの説明文といったアイテムの詳細の記入や，スクリーンショットやアイコン設定の画像, アセット，連絡先情報などを登録すると，マルウェアやウイルスなどの感染を機械的にチェックし，Google Play 上に公開される．

アプリケーションによっては，特定の機能を実現するためにアクセス権限を求める場合がある．アクセス権限とは，アプリケーションがユーザー側の端末のどの情報/機能を利用したいかを表現する情報である．アクセス権限には，ノーマルパーミッションとデンジャラスパーミッションの 2 種類があり，約 160 種の権限が存在する．ノーマルパーミッションは，アプリケーションがサンドボックス外のデータやリソースにアクセスする必要があるものの，個人情報や他のアプリケーションの操作に影響を及ぼすリスクがほとんどないケースが対象となり, ユーザの承認を必要としない．一方，デンジャラスパーミッションは，脅威をもたらす可能性があるものとして，特に重要な権限とされている．9 種類のデンジャラスパーミッションが存在し，それらの権限を利用するアプリケーションはダウンロードする際にユーザーの承認を求めている．

2.2 悪意を持つアプリケーションを検出するための研究

[2] では, Google Play から提供されている無料 Android アプリを対象として，アプリケーションの APK ファイルで記述されているパーミッションを抽出し，アプリケーション内で使用しているパーミッションからアプリケーションが属するべきカテゴリを推測する形で機械学習を行った．アプリケーションのカテゴリとアクセス権限は，密接な関係があるとし，同種のカテゴリと異なる特徴を持つアプリケーションは悪意を持ったものである可能性が高いと判断している．安藤ら [3] は，アプリケーションの紹介ページから情報を得て特徴量とし，機械学習の材料とすることで，悪意をもつアプリケーションの検出の精度が向上するかを確認した．評価実験の結果からは，あらかじめジャンル毎にアプリを分けてから悪意をもつであろうアプリケーションを検出するという手法をとることで，[2] のアプローチより精度の高い検出を得ることができていた．竹村ら [4]，林ら [5] は提案した手法を実現するためにデータの入手をし，そこから必要な情報を入手する仕組みを提案した．安藤ら [3] では，データセットの構築までは想定しておらず，集める特徴量を後から増やそうとした時に，すでに消去されているアプリケーションの情報が得られないなど，継続した研究を行うのに不十分な環境であった．

2.3 アプリ情報取得の手順

図 1 は竹村ら [4]，林ら [5] で提案された，アプリ情報取得システムの概要であり，表 1 は，この仕組みによって抽出される情報を表している．上部の図は各アプリケーションが利用する権限を取得する手順 (手順 A1〜手順 A5)，下部の図は各アプリケーションの紹介ページから抽出した情報を取得する手順 (手順 B1〜手順 B5) を表している．

手順 A1　各アプリケーションの APK ファイルを取得する．

手順 A2　入手した APK ファイルを APK ツールを使

用して展開し androidmanifest.xml を取り出す．各 APK ファイル内に存在するファイルを区別し管理できるよう保存する．

手順 A3　一部のバイナリ形式の部分の除去を行う．

手順 A4　実行時に要求する権限である「permission」「uses-permission」「uses-permission-sdk23」を抽出する．

手順 A5　各アプリケーションから抽出した情報を読み込み，権限の有無を記した表を作成する．

手順 B1　あらかじめ決めた取得する項目にしたがって，ダウンロード対象となる項目の URL を特定する．

手順 B2　対応する URL を指定してダウンロードを行うバッチファイルを作成する．

手順 B3　ダウンロード対象を定期的に入手する．アプリが削除されていないかも合わせて調査する．

手順 B4　入手したデータを分析し，抽出する情報を特徴量として抽出する．

手順 B5　アプリケーションごとの特徴量をまとめ表にする．対象アプリケーションが消されたかも表にする．

表 1　アプリ情報から入手した特徴量の一覧

項目	抽出する内容	抽出
アプリごとの各 APK ファイル 手順 A	アプリのダウンロードを行った後に各アプリの APK ファイルを取得	○
	展開した APK ファイル内から androidmanifest.xml を取り出す	
Google Play での紹介ページ 手順 B	カテゴリ	
	レビュー数	
	レビュー評価の平均	
	住所，メールアドレスが記載されているか	
	「Web サイトにアクセス」の有無	
	開発者が他にアプリを提供している数	
iOS 版の紹介ページ 手順 B	App Store に存在するか	
	レビュー数	
	レビュー評価の平均	
	開発者が他にアプリを提供している数	
アプリの公式サイト 手順 B	公式サイトの有無	
	最新情報の更新日	
	よくある質問の有無	
	公式 SNS の有無	
	公式 SNS の投稿数	
	公式 SNS の更新日	
アプリ名で検索した結果 手順 B	検索件数	
	検索結果の上位 3 つが関連しているか	
	関連キーワードの数	

2.4　Web スクレイピング技術について

Web スクレイピングとは，収集した Web サイトから自動的に情報を抽出するための技術で，ウェブ・クローラーあるいはウェブ・スパイダーとも呼ばれる．Web スクレイピングでは，Web 上で公開されている非構造化データを構造化されたデータに変換することにより焦点があてられており，ニュースポータル, ブログ, 財務報告などの Web 上で公開されているデータから，ニュースタイトルや更新日，特定の財務指標などのユーザが必要とするデータのみを抽出することを可能としている．Xpath とは，XML 形式の文書から特定の部分を指定して抽出するための簡潔な構文である．Web ページの情報を取得する際に XPath を用いて関数を変更し、文章をツリーとして捉えることで要素や属性の位置を指定することで，取得するデータの位置の特定を正確に行える．

3　評価用データセットの構築に関する考察

3.1　動機

竹村ら [4]，林ら [5] で実現している仕組みは，利用権限や公開されている情報をもとに，今後消される可能性のあるアプリケーションを事前に検出することができるなど，ソフトウェアの利用における注意喚起の支援を行う手法の基盤を構築することを目的としている．しかし実際のデータ収集はうまくいかなかった．竹村ら [4]，林ら [5] で行われたデータ収集の問題点として，大きく 2 つあると考えた．一つは、対象となるアプリケーションのリストを作成してから，実際に情報を取り出すまでに時間がかかりすぎてしまう点である．実際に必要な情報や抽出の方法を検討しながらデータ収集を行っていたので，その間に大きなタイムラグがあり，問題となるアプリケーションの情報が消されてしまい実際の収集までうまくいかなかった．もう一つは，データ収集の自動化の点である．表 1 の「○」で示している．利用権限の抽出のみ自動化の検討がなされたが，その他の抽出の自動化はできておらず，検討する必要がある．自動化を考慮した上で，竹村ら [4]，林ら [5] の仕組みの下でデータセットの構築を試み，提案された手法でデータ収集が可能であることを示す．

3.2　アプローチ

過去の研究で提案されたデータ収集のアプローチに従って，実際にデータ収集を行い，収集手法の実現可能性を確認する．次にデータ収集の自動化について考察を行う．収集したデータから情報を抽出する際にどれくらいの情報が自動的に収集可能となりそうかを確認する．

4　データセットの作成と構築について

4.1　データセットの構築方法について

竹村ら [4]，林ら [5] で作成したアプリ情報取得システムの仕組みを用いて，評価用データセットの構築を試みた．全体の手順を以下のように変更して行った．

手順 1　プロジェクトのリストアップを行う．

手順 2　必要なデータをどのように抽出するか確認を行い，APK ファイルの場所と特徴量を抽出するために必要な URL 群を特定する．

手順 3　権限の抽出について

A1　Android 端末を用いて，各アプリの APK ファイルを取得する．

A2　APK ファイルを展開し androidmanifest.xml を取り出す．APKtool を用いることで，バイナ

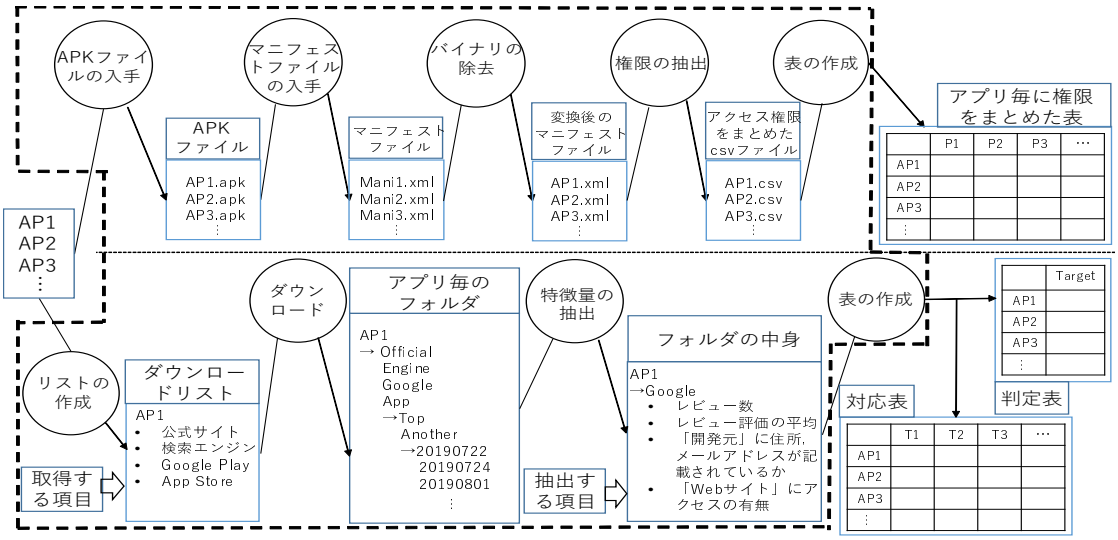

図 1　アプリ情報取得システム

リの除去も行える．

A3　抽出対象の権限である「permission」「usesper-mission」「uses-permission-sdk23」を抽出する．

A4　各アプリから抽出した情報を読み込み，権限の有無を記した表を作成する．

手順 4　アプリ紹介ページからの抽出について

B1　あらかじめ決めた取得する項目に従い，アプリごとにダウンロード対象の URL を特定する．

B2　対応する URL を指定してダウンロードを行うバッチファイルを作成する

B3　バッチファイルを起動し，ダウンロード対象を定期的に入手する．

B4　入手したデータを分析し，抽出する情報を特徴量として抽出し，表を作成する．

手順 5　ある一定期間過ぎた後に削除されたアプリの調査を行う．

4.2　データセットの収集について

データ収集を 7 ジャンルに対して行った．最初に，2020 年 7 月から 8 月の間に 1 ジャンルずつ 1 回目のリストアップを行い，リストアップ後に web ページの入手と APK ファイルの取得を行った．ジャンル毎に行うことで，一覧作成からデータ収集までの時間を短くし，アプリが消されてしまう問題を解消する．一度入手したデータについては数カ月にもう一度調査を行い，消されたかどうかの判断を行う必要があるが，この作業については特に問題となることがないと考える．実際には，データ件数を増やすために，定期的にリストアップを行い，新しいプロジェクトをデータセットに数カ月に一度追加していくことが必要であると考えられる．

データ収集を行った結果を表 2 示す．ジャンル毎に，1 回目のリストアップの日時，データの入手日時と（　）内でその件数を示している．リストアップから 3 日程度であれば，どのアプリケーションの情報も消されずに残っていた．新しく追加されたプロジェクトに対する調査を行った結果も，表 2 に示す．どのジャンルについても 40〜100 件程度のアプリが試行ごとに新規に追加されており，サンプルの入手には，定期的な入手の試行が必要であると考えられる．結果として 7 ジャンル計 1937 個のアプリケーションについてデータの収集が出来た．また，そのうち 476 個のアプリケーションは追加試行によるものであった．消されたアプリ数では，一定期間過ぎたアプリケーションの URL の有無の数を示したものである．アプリケーションのそれぞれのジャンルによって消されたアプリ数にばらつきが生じ，特に出会い系と音楽のジャンルに多く消されたアプリケーションが見られた．

表 2　7 ジャンルのアプリ群

アプリ群名	一覧作成日	データ入手日	追加データ入手日	入手できたアプリ数	消されたアプリ数
出会い系	8/29	8/29(246)	11/4(262) → 11/10(293) → 12/7(359)	359	20
音楽	8/6	8/9(300)	11/4(313) → 11/21(344)	344	23
ニュース	8/26	8/26(100)	11/10(140) → 12/7(173)	173	1
カレンダー	8/19	8/20(214)	11/10(233) → 12/7(265)	265	5
漫画	7/1	7/3(239)	11/10(266) → 12/7(314)	314	18
カメラ	7/15	7/17(150)	11/4(188) → 11/21(211)	211	8
ショッピング	8/26	8/27(212)	11/10(240) → 12/7(271)	271	3

5　情報の自動抽出に関する考察

5.1　抽出の自動化について

アプリを紹介する web ページからデータを抽出する手法として web スクレイピングの技術を用いる方法を試した．具体的には，web スクレイピングサービスを提供する Octoparse[6] に対して，各アプリを保持する URL を提供し，データの自動抽出を試みた．

表 3 アプリ情報から入手した特徴量の一覧

項目	抽出する内容	抽出	自動抽出	Octoparse に提供した URL
アプリごとの各 APK ファイル	アプリのダウンロードを行った後に各アプリの APK ファイルを取得	○		
	展開した APK ファイル内から androidmanifest.xml を取り出す			
Google Play での紹介ページ	カテゴリ		◎	紹介ページの URL
	レビュー数		◎	紹介ページの URL
	レビュー評価の平均		◎	紹介ページの URL
	住所，メールアドレスが記載されているか		◎	紹介ページの URL
	「Web サイトにアクセス」の有無		◎	紹介ページの URL
	開発者が他にアプリを提供している数		◎	提供アプリ一覧の URL
iOS 版の紹介ページ	App Store に存在するか		−	リスト作成時に抽出
	レビュー数		◎	紹介ページの URL
	レビュー評価の平均		◎	紹介ページの URL
	開発者が他にアプリを提供している数		◎	提供アプリ一覧の URL
アプリの公式サイト	公式サイトの有無		−	リスト作成時に抽出
	最新情報の更新日			
	よくある質問の有無			
	公式 SNS の有無		−	リスト作成時に抽出
	公式 SNS の投稿数			
	公式 SNS の更新日			
アプリ名で検索した結果	検索件数		◎	Octoparse で抽出
	検索結果の上位 3 つが関連しているか			Web から直接抽出 (判別不可)
	関連キーワードの数		◎	検索ページから Octoparse で抽出

5.2 抽出可能な情報について

表 3 は表 1 の項目でそれぞれスクレイピングを行い，自動抽出できたものに「◎」を付けた表である．それぞれの項目で自動抽出できたものに，どのように抽出できたか記述した表である．Web スクレイピング技術は，アプリケーションの紹介ページや各通販サイトのように，同じ構造である限り，各ページから共通性のあるデータを抽出することが出来る．Googleplay，iTunesStore，検索結果などは同一の形式で表現されるので，URL を与えることで自動抽出が十分可能であることが分かる．

5.3 抽出のための Xpath の記述について

Xpath は HTML などの文書を特定するための技術であり，データを抽出する際に特定の要素位置を，設定できれば簡単に抽出ができる．例えば，Octoparse では Xpath を属性で指定しており，タグに属性をつけることで要素の効果を指定したり，具体的な指示を付け加えることができる．例として，抽出された Xpath，/descendant-or-self::DIV[@class=” KoLSrc”] をあげると，属性について class のような要素に紐づく属性を Xpath では@で表し，抽象化すると，Xpath 構文は//タグ名 [@属性名=” 属性値”] と表すことが出来る．

6 まとめ

本研究では，竹村ら [4]，林ら [5] によって提案された仕組みに基づいて実際にデータ収集を行った．リストアップからデータ収集までの時間を短くすることで十分実現可能で，繰り返しの試行で件数を増やすことができることを確認した．データ抽出の自動化では，Web スクレイピング技術を用いて 3 でのほとんどの項目についてデータ収集出来た．自動化が可能になった項目が増えたことで，データ収集を行う時間の短縮が可能になり，更に多くのデータ収集が見込める．

今後の課題は，ジャンルにおけるアプリケーション一覧が入手ができそうかについて本当に抽出が可能かどうか確かめる．また，入手するデータ数を多くすることと，集めたデータから今後消される可能性のあるアプリケーションの検出を得られた特徴量から機械学習を行うことである．

7 参考文献

参考文献

[1] Google play：https://play.google.com/store/

[2] Zhongmin Ma：“Android Application Install-time Permission Validation and Run-time Malicious Pattern Detection”，Master thesis of Virginia Polytechnic Institute and State University，2013.

[3] 安藤花風里，伊藤美惟：“脅威を引き起こすアプリケーションを アクセス権限などを用いて検出する手法についての考察 ”，南山大学理工学部 2018 年度卒業論文，2019.

[4] 竹村大樹，駒田涼，真野航平：“脅威を引き起こすアプリケーションを アクセス権限などを用いて検出する手法についての考察 ”，南山大学理工学部 2019 年度卒業論文，2020.

[5] 林勢也，川村隼太：“脅威を引き起こすアプリケーションを アクセス権限などを用いて検出する手法についての考察 ”，南山大学理工学部 2019 年度卒業論文，2020.

[6] Octoparse：https://www.octoparse.jp/tutorial/xpath/