脅威を引き起こすアプリケーションを検出する手法についての考察
ーアクセス権限の抽出方法についての考察ー

2015SE080 竹村大樹　2016SE042 駒田涼　2016SE050 真野航平
指導教員：横森励士

1　はじめに

近年のスマートフォンの普及により，スマートフォン上で動作するアプリケーション（以下，アプリ）の需要が増加している．大量のアプリの中には悪意を持つものも多く存在しており，様々な脅威が日々引き起こされている．このような環境下では，アプリの紹介ページなどの利用者に事前に公開される情報を利用して，悪意を持ったアプリによって引き起こされる脅威を回避することが求められる．[1] では，アプリが利用する権限や紹介ページなどから入手可能な情報を用いて機械学習を行うことにより，高い精度で悪意をもつアプリの検出ができそうであることを示した．しかし，情報をアプリの紹介ページから直接得ていたので，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．

本研究では，特徴量を抽出する際に必要な情報を手元に残したうえで，手元のデータから特徴量を抽出するような仕組みを作ることを目的とする．実際には，アプリの APK ファイルから利用権限情報を抽出し，機械学習のための表を作成する．特徴量を得るためのシステムを試作し，実際に公開されているアプリに対して入手を試みた結果を紹介するとともに，それらの入手したデータと得られた特徴量をもとにいくつかの教師ありの機械学習手法を適用した結果を紹介する．これらの結果を通じて，現状の入手手法の課題と，得られた情報を用いて機械学習を行うに際して直面するであろう課題を考察し，悪意を持つアプリの検出を行う仕組みの実現につなげる．

2　背景技術

2.1　悪意を持ったアプリが引き起こす脅威

代表的な Android アプリの配布サービスとして，GooglePlay が運用されている．アプリを提供する側がアプリ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た上で，Google Play 上に公開される．チェックを通り抜ければ，悪意を持つアプリもそのまま公開されてしまうので，利用者はアプリをインストールする際に Google Play から与えられた情報をもとに自己判断を行う必要がある．悪意を持つアプリが引き起こす問題の例としては，個人情報抜き取り，端末の遠隔操作，悪意を持った Web サイトへの誘導などがある．Android 不正アプリ検出数の割合 [2] を表 1 に示す．表 1 で示す通り，“アドウェア”が約 8 割を占めており，“情報窃盗／バックドア”が残りの部分の半数を占めている．ユーザ側はこれらの脅威への対策としてセキュリティアプリを入れることが推奨されているが，そのどれもが一度アプリをインストールしてからチェックにかける方式をとっているので，インストールされた時点で何らかの被害を及ぼすアプリには効果が薄いと言える．

表 1　国内での不正アプリ検出種別割合 (2015) [2]

脅威の種類	割合
アドウェア	79.80 %
情報窃盗/バックドア	8.56 %
ネット詐欺	2.84 %
脆弱性悪用	1.46 %
プレミアム SMS 悪用	0.81 %
ランサムウェア	0.04 %
その他の不正アプリ	6.48 %

2.2　関連研究

アプリのアクセス権限に対して，機械学習による分類分けを行い，悪意を持ったアプリの判別を行った Zhongmin らによる研究 [3] がある．[3] 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た．アプリ内で使用しているパーミッションからアプリ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．

安藤らによる研究 [1] では，パーミッションに加えてアプリ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，悪質なアプリ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持つであろうアプリを検出するという手法をとることで，[3] 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．

2.3　権限について

アクセス権限とは，アプリ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 種類があり累計約 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，ユーザーの承認を必要としない．デンジャラスパーミッションとして，9 個の権限が脅威をもたらす可能性があると認定され，特に重要な権限とされている．それらの権限を利用するアプリはダウンロードする際にユーザーの承認を求めている．

3　アクセス権限の抽出方法について

[1] では，アクセス権限や紹介ページからの情報を用いて機械学習を行うことで，より高い精度で悪意をもつアプリの検出ができそうであることを示した．情報収集の観点からは，権限を含めたそれらの情報は，紹介ページを確認することで得ており，後から特徴量を増やそうとした時に，既に消去されたアプリは検出に必要な情報を得ることができない．特徴量の抽出に必要な情報は手元のデータから抽出するような仕組みが分析環境の実現に必要である．

本研究では，抽出された情報から悪意のあるアプリを検出するシステムの構築を行う．システムでは事前に必要となりそうな情報の範囲を考察し，それらの情報を全て保存しておく．例えばアクセス権限については，対象となる Android アプリのアーカイブである APK ファイルを保存しておく．それぞれの調査項目を抽出する際には，それらの情報から抽出を行い，各アプリについて，利用する権限や，紹介ページの情報についての調査項目を表にまとめ，その表を用いて機械学習を行う．以下では，システムの全体像と自分たちが担当した範囲について紹介を行うとともに，得られた情報をいくつかの機械学習手法に適用し，どのような結果が得られたかについて紹介する．

4　アプリの特徴量取得システム

4.1　システム全体の概要

図 1 はアプリ情報取得システムの概要で，悪意のある Android アプリの検出を目的とする．各アプリから 2 種類の表を作成し，その表を用いてアプリが悪意を持つかどうかを判断する．1 つ目の表は，各アプリが利用する権限をまとめた表で，2 つ目の表は，各アプリの紹介ページなどから抽出した情報をまとめた表である．アプリが悪意を持つかの判断は，アプリの紹介ページから得られた情報から判断し，アプリ紹介文で矛盾があるもの，セキュリティアプリで検出されたもの，配信が停止されたものを悪意があったアプリとみなした．

本研究では図 1 の上部に相当する，各アプリの APK ファイルから抽出した利用権限に関する情報をまとめる部分を担当し，抽出したデータを機械学習の材料とする．

4.2　利用権限抽出部の概要について

図 2 は，アプリの APK ファイルから得られる情報から利用権限を抽出するサブシステムの概要である．APK ファイルには，androidmanifest.xml が含まれており，パッケージ名の指定，コンポーネントの記述，パーミッションの宣言，アプリ実行時情報 (Instrumentation) の記述，必

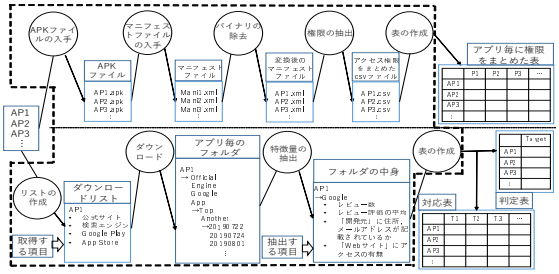

図 1　アプリ情報取得システム

要な API レベルの記述，必要なライブラリの記載などを行っている．本研究では，パーミッションの宣言について着目し，具体的な記述要素として「permission」,「uses-permission」,「uses-permission-sdk23」の要素を抽出する．

4.3　利用権限の取得手順

想定する入力は，同じジャンルのアプリの集合についての情報で，AP ＝｛AP1，AP2，AP3，・・・APK｝と表す．想定する表はそのアプリの集合において出現した権限の集合 P ＝｛P1，P2，P3，・・・PK｝について AP × P を表す表となる．以下に抽出の手順を示す．

準備

同じジャンルのアプリについてのアプリの情報を収集し，アプリの集合 AP を作成する．

手順 1:APK ファイルの入手

それぞれのアプリごとに APK ファイルを入手する．Android 端末を用いて，アプリのダウンロードを行った後に，各アプリの APK ファイルを取得する．

手順 2:APK ファイルの展開

APK ファイルを展開し，中身を取り出す．

手順 3:マニフェストファイルの取出し

展開した APK ファイル内から，目的となるファイルである androidminifest.xml を取り出す．それぞれの APK ファイル内に存在するファイルを区別して，管理できるように「AP1.xml」のように保存する．

手順 4:バイナリ部分の除去

入手した androidmanifest.xml は一部がバイナリ形式なので，バイナリ部分について除去を行う．

手順 5:アクセス権限の抽出

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 6:表にまとめる

各アプリから抽出した情報を読み込んで，アプリ毎に各権限の有無を 0，1 で表現した表を作成する．

5　データセットの作成と機械学習の適用結果

5.1　機械学習を行うためのデータセットの作成

前章で作成した権限取得システムをもとに，表 2 で示す 4 ジャンルの計 426 個のアプリからなるデータセットを

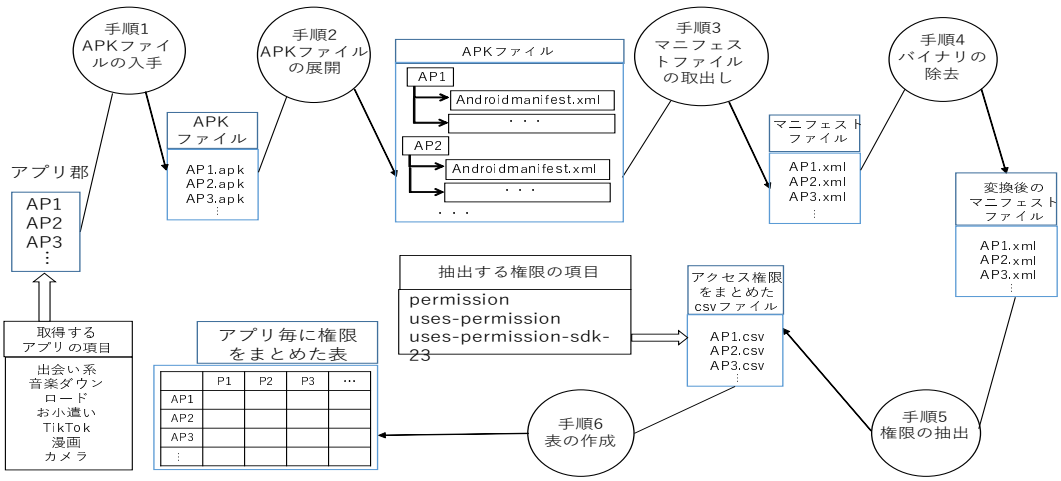

図 2　権限取得システム

0，1 の表で作成した．Android が用意している利用権限である約 160 種の利用権限を含む，計 355 種のアクセス権限が得られた．アプリがその利用権限を持っていれば 1 で，持っていなければ 0 で表す．また悪意のあるアプリを 1，悪意のないアプリを 0 とした表も準備した．

2019 年 7 月に対象アプリをリストアップし，2019 年 8 月〜11 月の間に対象アプリの APK ファイルを逐次入手し，利用権限の抽出を行った．抽出の方法の決定と並行しながら APK ファイルの入手を行っていたので，ジャンルによっては入手の時期が遅く，分析対象から外したジャンルも存在した．実際に機械学習を行おうとした際にサンプル数不足が生じたので，ジャンル情報も特徴量とした 1 つの集合としても機械学習を行った．

表 2　4 ジャンルのアプリ群

アプリ群名	サンプル数	悪質なアプリ数
出会い系	125	4
TikTok	86	21
お小遣い	115	3
漫画	100	0

5.2　アプリ情報から入手した特徴量による分析結果

表 2 のデータセットに対して k-最近傍法，線形モデル，ランダムフォレスト，勾配ブースティング決定木，サポートベクターマシンの 5 つの教師あり学習手法を適用した．訓練セットとテストセットを 3 : 1 に分割したのちに機械学習を行う試行を 5 回ずつ行った．各試行において，検出のパラメータ（しきい値）を変えることで，再現率や偽陽性率がどう変化するかをまとめ，グラフ上で表現した．

実際に分析を行ったところ，k-最近傍法では悪意のあるアプリが多いジャンルでは良い結果となった．ランダムフォレスト，勾配ブースティング決定木では，良い結果となったが，検出のパラメータ（しきい値）が 3 点しか存在しない．悪質なアプリの数が少なく，まだ十分な機械学習の環境でないと考えられる．サポートベクタマシンによる分析結果も極端な結果となっている．ノーマルパーミッションも含めることで，悪質なアプリが実は必要な機能を実現していなかったなどの理由で，検出が容易になったということも考えられるが，悪質なアプリの数が少なく，十分な機械学習の環境でないからかもしれない．線形回帰で分析を行った結果を図 3 に示す．ジャンルをまとめることで，偽陽性率が 0.2〜0.4 くらいのある程度低い値の中で，一定以上の再現率を実現できていた．ジャンル分けした結果のほうが良い結果になりやすい傾向はあるが，ジャンル分けした結果においては安定して機械学習が行えておらず，悪質なアプリに関する情報がまだ十分でないと考えられる．

5.3　線形回帰における各権限の重みについて

線形回帰モデルで機械学習を行った場合，それぞれの権限に対して重みが付けられ，そのモデルにおいて悪意を持つアプリかどうかを判断するための材料となっている．今回 5 回の試行における重み付けの傾向についても調査を行った．悪意を持つアプリであることの判断材料となった権限は，「システムの起動時にアプリを起動する」，「フォアグラウンドサービスを起動する」，「バックグラウンド上で作業を行う」，「連絡先の読み書きをする」などを行うための権限であった．一方で，「インターネットに接続する」，「ネットワーク接続に関する情報を入手する」などの一般的な権限を要求するアプリは安全であると判断される傾向にあった．

6　考察

特徴量を入手するシステムを用いて，利用権限に関する情報を入手する仕組みは実現できたが，さらなる自動化が必要であると考えられる．例えば，利用権限の取得にある手順 1 の APK ファイルの入手において Android 端末を使用していたが，Android 端末から APK ファイルを送信できなかったアプリが存在した．このことからリストアッ

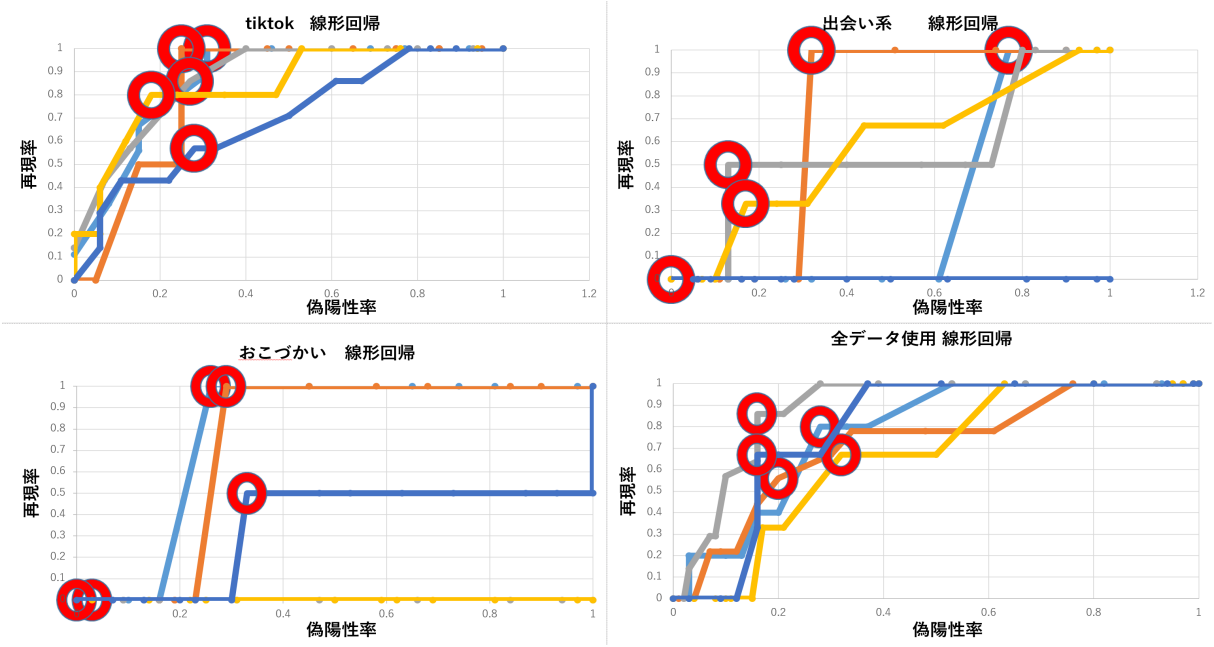

図 3　線形回帰での偽陽性率-再現率のグラフ

プはしたが，アプリ集合に含めることが出来ないアプリも存在した．このような原因で，実際に分析対象となる悪意を持つアプリの抽出を一部行えなかった．対策として，Chrome の拡張機能を用いることで APK ファイルを取得することが出来た．しかし，手動で入力する必要があるので，効率的に大量の APK ファイルを取得する方法を確立する必要がある．さらに，リストと実際に取得した APK ファイルに差異があるなど，アプリ情報との分析結果の連携が十分でなかった．アクセス権限とアプリ情報を，常に最新の状態に保つことが必要である．機械学習を行った結果を確認すると，ジャンルごとに分けて分類した結果のほうが，より良い機械学習結果が得られる傾向があるが，安定して機械学習が行えているわけではなく，悪質なアプリに関する情報がまだ十分でないと考えられる．そのため，悪質なアプリのデータの件数を増やす必要がある．リストを作成してから，実際に APK ファイルを収集するまでの時間が長かったことで，悪質なアプリは APK ファイルの取得前に消されてしまっていた．悪質なアプリのデータ件数不足となった原因であり，そのため，リスト作成と APK ファイルの取得を同時に行うという作業を時間をおかずに実行したうえで，何回も繰り返すという対策が必要である．線形回帰のみ結果を示すことが出来た．データセットの中には，1 つのアプリでしか権限の定義を行っていないものが多数存在し，必要のない情報が多く含まれていたと考える．現状では権限の選別を行っていないので，データセットを最適化し，余分なデータを削除する必要がある．

線形回帰における各権限の重みについて調査したところ，傾向として悪質なアプリが要求しやすい権限がありそうだということも分かった．判断材料となっていた権限はほぼノーマルパーミッションで，デンジャラスパーミッションには含まれてなかった．それらの 9 権限を重要な権限として扱いを変えるべきであるという提言を行うことができる可能性がある．

7　おわりに

本研究では，特徴量を抽出する際に必要な情報を手元に残した上で，手元のデータから特徴量を抽出する仕組みを作成した．実際のアプリに対して適用を行い，入手可能であることと，アクセス権限が [1] と同じ傾向の結果であることを確認したが，安定した結果が出ておらず，悪質なアプリに関する情報をさらに入手する必要があることを確認した．機械学習の精度の向上を目的として，アプリ情報を定期的に入手し，データセットを拡充することが今後の課題である．

参考文献

[1] 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 2018 年度卒業論文，2019.

[2] トレンドマイクロ：1000 万個を突破した Android 不正アプリの「これから」，http://blog.trendmicro.co.jp/archives/12960．2016.

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