必要とするアクセス権限に基づく Android アプリケーション分類手法の提案

2014SE028 石田尚也　2014SE052 小林薫　2014SE082 大野哲弥
指導教員：横森励士

1　はじめに

近年 Android を筆頭に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．それに伴い，大量のアプリの中には悪意を持つものが存在しており，それらによって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，アクセス権限など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未然に回避することが求められる．

本研究では，同じような機能を持ったアプリの集合に対し，アクセス権限の有無で階層クラスター分析による分類を行う手法を提案する．この手法を用いて，他のアプリとは要求する権限に大きな差があるアプリを分類し，悪意を持つアプリの検出が可能かを検討する．提案手法を用いることで，悪意を持つアプリを検出することができれば，これらの被害を未然に防ぐことができるのではないかと考える．

2　背景技術

2.1　Android アプリの配布サービスと悪意を持ったアプリが引き起こす脅威

Android アプリの配布サービスとして，Google Play[1] が提供されている．アプリを提供する側がアプリに関する情報を登録すると，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．その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．そのため，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を推奨されているが，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受ける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　アクセス権限

アプリをインストールする際に Google Play はアプリが端末のどの機能や情報にアクセスするかの情報を利用者に確認させる．権限は大きく分けて 17 のカテゴリ [3] に分けることができ，表 2 のようにそれぞれの用途ごとにアクセス権限を利用者に要求する．例えば，現在地周辺のマップを表示するアプリの際には，インストール時に「位置情報」の権限の使用が要求される．このアクセス権限には依存関係があり，例えば，「ID」と「連絡先」では「この端末上のアカウントの検索」という項目が重複している．また，アクセス権限には Dangerous パーミッションと呼ばれる権限グループがある．これを表 2 の“※”で表す．これらはアプリがユーザの個人情報を含むデータやリソースを必要とする，あるいはユーザが保存したデータや他のアプリ操作に影響を及ぼす可能性があるものを指し，ユーザがアプリをインストールする際に最も気を付けるべき権限グループである．尚，この Dangerous パーミッションには依存関係はない．

アドウェアの検出は，一般的に広告の内容を調査する必要があり，アクセス権限の有無から分類するのは困難である．一方で情報窃盗やバックドアでは，悪意を持つアプリが脅威を引き起こすためには，利用者がアクセス権限の許可を与えることが必要である．本研究ではそれらのアクセス権限を利用する機能を含むアプリが提案手法でどのように分類されるかを評価する．

2.3　関連研究

Teufl らによる研究 [4] では大量のアプリを Google Play のカテゴリとアクセス権限を用いてクラスター分析を行い，分類分けされた結果の関連性を示した．分類結果では，『アカウントデータにアクセスできるアプリ』,『位置情報を利用するアプリ』,『端末がスリープになることを防ぐような音楽や動画のアプリ』など，要求するアクセス権限に典型的なパターンが存在し，クラスター分析によってそれらに分けることができたことが示されている．

一方で，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

表 2 アクセス権限の一覧 [3] より引用

アクセス権限	用途	アクセス権限	用途
アプリ内購入	利用者に対して アプリ内購入を求める	※ストレージ	端末のファイルや保存されている データの使用を許可する
※カメラ	端末のカメラの使用を許可する	※マイク	端末のマイクの使用を許可する
ID	端末のアカウント情報や プロフィール情報の使用を許可する	※位置情報	端末の位置情報の使用を 許可する
※連絡先	端末の連絡先情報の使用を許可する	Wi-Fi 接続情報	端末の Wi-Fi 接続情報の 使用を許可する
※カレンダー	端末のカレンダーの情報の 使用を許可する	※電話	電話やその通話履歴の 使用を許可する
※ SMS	端末のテキストメッセージや マルチメディアメッセージの サービスの使用を許可する	端末 ID と 通話情報	端末の ID，電話番号， 電話中かどうかの情報， 通話相手の番号にアクセスを許可する
端末とアプリの履歴	機密ログデータや システム内部の状態，ウェブの履歴， 実行中のアプリの取得を許可する	モバイルデータ通信 の設定	モバイルデータ接続と， 受信するデータをコントロールする 設定の使用を許可する
画像/メディア/ ファイル	端末のファイルや保存されている データの使用を許可する	※ボディセンサー	心拍数モニターなどウェアラブル センサーへのアクセスを許可する
その他	端末メーカーが指定した カスタム設定や，アプリ固有の権限の 使用を許可する（例:Bluetooth など）		

た Zhongmin らによる研究 [5] がある．[5] 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機械学習に使用して，アプリのカテゴリごとにしきい値を計算し，得られたしきい値からカテゴリ内アプリを良性か悪性かに分類分けを行った．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ものは悪意を持ったものである可能性が高いとしている．

3 分類手法

3.1 必要とするアクセス権限に基づく Android アプリ分類手法の提案

[5] で示すように，悪意を持つアプリは本来必要となる権限とは別に余分な権限が必要であることが想定される．また，[4] で示されているように，アクセス権限に基づいてクラスター分析を行うことで，要求するアクセス権限が似たアプリ群に分類できる．同種のアプリに対して私たちが分類を行ったところ，そのような同種のアプリに対する分類でも，その中で同じような権限を要求するアプリにわかれることが分かった．そこで，同じような権限を要求するアプリに分類した上で，他のアプリと異なるアクセス権限を要求するアプリが仲間外れになるように分類できれば，多大なコストをかけることなく，効率的に悪意を持つアプリが検出できると考えた．

3.2 アクセス権限を表現するベクトル

提案手法では，あるアプリ A におけるアクセス権限 j に基づいて 2 種類のベクトルを定義する．1 つ目のベクトルは，表 2に示す 17 カテゴリの権限の重みを等しくすることで，似た権限を要求するアプリを分類するために利用する．このベクトル$\vec{V_a}(A)$ を

$$\vec{V_a}(A) = (a_1,\ a_2,\ \cdots,\ a_j,\ \cdots,\ a_{17})$$

と定義する．あるアプリ A がアクセス権限 j を有しているのであれば $a_j = 1$ とし，j を有していないのであれば $a_j = 0$ とする．

2 つ目のベクトルは，アクセス権限の出現頻度によって重み付けを行うことで，利用している権限が特殊なものを仲間外れにするために利用する．ここでは Dangerous パーミッションでベクトルを作成し，脅威と関連の強いアクセス権限のみを考える．このベクトル $\vec{V_b}(A)$ を

$$\vec{V_b}(A) = (b_1,\ b_2,\ \cdots,\ b_j,\ \cdots,\ b_9)$$

と定義する．あるアプリ群内において重み付けを行う時，群内のアプリ数を N，群内でアクセス権限 j を要求するアプリ数を $Sum(j)$ とし，以下のように b_jを定義する．

$$b'_j = \begin{cases} \frac{N}{Sum(j)} & \text{権限を必要とする} \\ 0 & \text{権限を必要としない} \end{cases}$$

3.3 分析の手順

1. 分類対象のジャンルのアプリを入手する．
2. アプリ毎にアクセス権限を調査し，それぞれについて V_aを定義する．
3. $\vec{V_a}$を入力として，階層クラスター分析を行い，樹形図を得る．樹形図における結果から，似た権限を必要とするアプリのグループを抽出する．
4. グループ内のアプリ毎にアクセス権限を調査し，それぞれについて V_bを定義する．
5. $\vec{V_b}$を入力として，階層クラスター分析を行い，樹形図を得る．得られた樹形図で距離が離れたアプリについて分析を行う．

表 3 調査対象のアプリ群一覧 (アプリ群名 | サンプル数)

動画プレイヤー	100	便乗系 (計 7 種)	494	レシピ	48	2 チャンネル	30
出会い系	100	コミック	30	学習	30	ライト	30
節電	58	計算機	55	ダイエット	30	翻訳	50
バッテリー	28	エミュレーター	46	広告ブロック	50	無音カメラ	40
アニメ見放題	30	クーポン	40	電話帳	30	バイト検索	40
クリーナー	30	SNS	34	お小遣い	60	アラーム	40

表 4 分析結果の一部 (適合率による順位付け)

順位	アプリ群名	サンプル数	仲間外れ	悪意持ち	正解数	適合率	再現率
1	計算機	55	2	5	2	1.0	0.4
2	お小遣い	60	7	9	6	0.86	0.67
3	エミュレータ	46	8	14	5	0.63	0.36
15	クリーナー	30	7	4	3	0.43	0.75
16	出会い系	100	31	56	13	0.42	0.23
28	翻訳	50	7	6	1	0.14	0.17
29	スプラトゥーン	69	9	16	1	0.11	0.06
30	コミック	30	3	1	0	0	0
-	平均	-	-	-	-	0.41	0.38

4 評価実験

4.1 アプリの調査

Android アプリを対象とし，表 3 のような 1532 個のアプリから構成される 30 のアプリ群を分析対象として評価実験を行った．実際のアプリ群に対して，提案手法によるクラスター分析を行い，アプリ群内のアプリがどのように分類されるかを調査した．目的を達成する上で，どのようなアプローチで分析を行うべきかについて考察を行い，悪意を持つアプリがどのように配置されるかについて調査した．評価においては，4 つの基準のいずれかを満たしたものを悪意を持つアプリであるとみなした．

レビューでの報告
ユーザの主観による意見ではなく，実際に起きた事象についての被害報告のみを考慮した．

セキュリティアプリでの検出
セキュリティアプリに搭載されているアプリスキャン機能によってマルウェアやプライバシー保護の観点から危険性があると判断されたものを対象とした．

アプリ紹介文の矛盾
紹介文においてアクセス権限の用途について触れていない場合や記述と矛盾するアクセス権限を要求しているものを対象とした．

アプリ配信の停止
Google Play 上において該当アプリがすでに削除されているものを対象とした．

4.2 適用例

提案手法による分類結果をサンプル数 100 の動画プレイヤーを例に用いて示す．$\vec{V}_a$を用いて似た権限を要求しているグループに分類した結果，図 1が得られた．私たちはこの図からアプリを 4 つのグループに分類し，調査を行った．左から 1 番目のグループは基礎となる権限を持たず，ストリーミング再生が主体のグループであった．2 番目のグループは「ストレージ」を基礎の権限とし，自分の端末に保存されている動画を再生するグループであった．3 番目のグループは「電話」と「端末 ID と通話情報」を基礎の権限とし，会員登録が必要となるグループであった．4 番目のグループは「ID」と「連絡先」を基礎の権限とし，アカウント作成が求められるグループであった．

次に$\vec{V}_b$を用いて，他と異なる権限を要求しているものを分離させるように分類した．例として，図 1の左から 2 番目のグループの分類結果を図 2に示す．この分類方法では他のアプリに比べてアクセス権限に大きな差があるものほど外側に配置され，図 2では外側から順に「カメラ①」,「電話②」,「マイク③」,「位置情報④」,「ストレージ⑤」の権限を要求していることが分かった．この図で 4.1 節の基準を満たしたものを赤枠で示すと，1 番目と 4 番目の集合に固まって現れた．私たちの研究では，図の外側に配置されたアプリがどれだけ赤枠で示されるかと，赤枠で示されたものが内側に配置され，埋もれていないかが重要となる．この例では，赤枠で示されたものが多く存在した 4 番目の集合が外側に配置されることが望ましかったが，「電話」と「マイク」の権限を要求しているものが少なかったため，「位置情報」の権限を要求しているものが内側に配置され埋もれてしまっていたことが分かった．

4.3 全体的な結果

表 4 に提案手法によって得られた結果の一部を示す．実験では，似た権限を要求しているアプリのグループ毎にしきい値を設定し，クラスター分析を行い，そのしきい値以下で生成されたアプリの集合のうち，集合に属するアプリ数がグループ全体の半分未満の場合，その集合に属するア

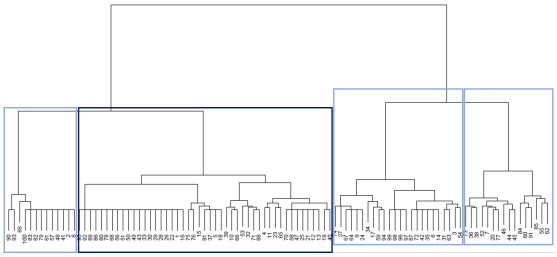

図 1　カテゴリ分け (動画プレイヤー)

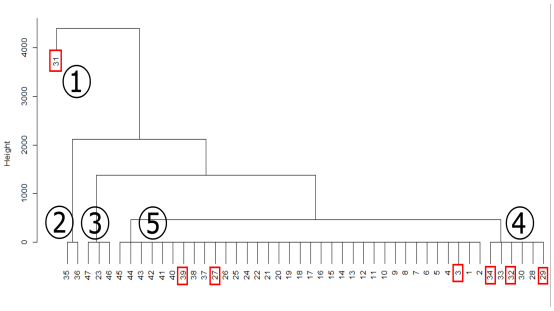

図 2　出現頻度でクラスター分析

プリを「仲間外れ」であるとみなした．グループ毎に，4.1 節の基準に基づいて悪意を持つものをあらかじめ抽出しておき，仲間外れになったアプリのうち，どれだけが悪意を持つものであったかを「適合率」，悪意を持つものをどれだけ仲間外れに出来たかを「再現率」として求めた．本実験では，仲間外れのアプリが悪意を持っているかに重点を置き，適合率によって順位を定めた．

順位が上位のものと下位のものの適合率の振れ幅が大きいことや平均値と中央値に差が見られないこと，再現率が低いことから，仲間外れにしたものが悪意を持っていると断定することは困難であり，同種なアプリと比べて要求しているアクセス権限に大きな差がみられたとしても悪意を持っているとは言えないことが分かった．また結果の分析を行っていく上で，4.1 節の基準を満たすアプリが，図 2 の 4 番目の集合のように比較的まとまって現れる現象が多くみられることが分かった．

5　考察

提案手法を用いてアプリを分類した場合，4.3 節で示した通り適合率は約 4 割に留まり，結果として，同種なアプリと比べて要求しているアクセス権限に大きな差がみられるからといって，悪意を持っているとは言えないことが分かった．結果について分析していく上で，悪意を持ったアプリが比較的まとまって現れる現象が多くみられたことが分かった．これに対して，私たちは悪意を持つアプリに利用する権限に共通性があることが原因であると考えた．動画プレイヤーでの事例では，悪意を持つアプリのほとんどが“位置情報”のアクセス権限を要求していた．これにより他のアプリとのアクセス権限の差が小さくなり，本来安全とみられるべき多機能型の動画プレイヤーのアプリが余分なアクセス権限を求めているとみなされてしまい，仲間外れになってしまっていた．この問題に対してソフトウェア工学の観点から考えると，ユニークな機能を持ったアプリを開発した際，実際に悪意を持ったアプリに比べてより危険なアプリに捉えられてしまうことが考えられる．そのためユニークな機能を実装する際には，アプリ紹介文などでアクセス権限の要求理由を明示しておくことが望ましい．また，この悪意を持ったアプリが利用している権限に共通性がみられる特徴を利用して，予め対象データに過去の不正アプリの事例からアクセス権限を付与した検出用のサンプルを追加して分析を行うことで，そのサンプルと同じ集合にまとまったアプリは悪意を持つアプリである可能性が高いのではないかと考えた．

6　まとめと今後の課題

本研究では，同じような機能を持ったアプリ群に対し，アクセス権限の有無による階層クラスター分析を行い，機能による細かい分類分けと，他と異なる権限を要求するアプリの検出を行った．評価実験からは権限に大きな差があるものが悪意を持つとは限らないこと，悪意を持ったアプリが利用する権限に共通性があることを確認した．共通性を考慮することで，提案手法の精度の向上が期待できる．不正アプリの事例を収集し，悪意を持ったアプリが要求するアクセス権限の共通性を把握することが今後の課題となる．

7　参考文献

参考文献

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

[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] Google ヘルプ：Android 6.0 以降のアプリの権限を管理する，https://support.google.com/googleplay/answer/6270602.

[4] P. Teufl，S. Kraxberger，C. Orthacker，G. Lackner，M. Gissing，A. Marsalek，J. Leibetseder，and O. Prevenhueber：”Android Market Analysis with Activation Patterns”，Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering，vol. 94，pp. 1-12，2011.

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