Seleção de Características Multiobjetivo para Detecção de *Malwares Android*

Philipe Fransozi
Jhonatan Geremias
Eduardo K. Viegas
Altair Santin

Pontifícia Universidade Católica do Paraná

PUCPR

CAPES

Motivação

- Considerando um conjunto de características, extraídas de um arquivo de aplicação *Android*, utilizado como um vetor de *features* em um modelo de classificação, existe algum subconjunto que melhore as taxas de erro e o tempo de inferência?

Tópicos

- **Introdução**
- Background
- Trabalhos Relacionados
- Proposta
- Resultados
- Conclusão
- Trabalhos Futuros
- Perguntas

Introdução

- **Contexto Acadêmico**: pesquisa foi realizada durante o PIBIC 2023-24, como resultado três artigos foram produzidos:
  - *Seleção de Características Multiobjetivo para Detecção de Malwares Android,* aceito na SBSeg 2024, categoria WTICG;
  - *APKAnalyzer: Ferramenta de Classificação de Malwares Android Baseada em Multi-view e Seleção de Características Multiobjetivo*, aceito na SBSeg 2024, categoria Salão de Ferramentas;
  - *A Multi-view Android Malware Detection Model Through Multi-objective Optimization*, submetido a ICMLA;

Introdução

- **Contexto dos dispositivos móveis Android**:
  - Em 2024, domínio do mercado com 3/4 de *marketshare*, aproximadamente 3 bilhões de usuários;
  - Em 2023, houve um aumento na detecção de aplicativos maliciosos, aproximadamente 53% de aumento; grande parte dessas amostras oriundas da loja oficial de aplicativos *Google Play*.

Introdução

- **Contexto dos métodos de análise de malware Android**:
  - **Análise dinâmica** obtém comportamentos maliciosos durante a execução do aplicativo. Trata-se de um método que exige maiores esforços relacionados ao ambiente para simulação (sandbox) e aos estímulos adequados para reprodução do comportamento malicioso;
  - **Análise estática** obtém indícios de um comportamento malicioso a partir do conteúdo do aplicativo (permissões obtidas do arquivo manisfest.xml). Trata-se de um método mais simples, rápida implementação e escalável.

Introdução

- **Contexto dos métodos de detecção de malware Android**:
  - Na literatura, encontramos inúmeros métodos para a tarefa de detecção, dentre os quais temos os métodos baseados em modelos de aprendizado de máquina. Em geral: um vetor de características é utilizado como entrada de um modelo de classificação que o classifica como malicioso ou benigno;
  - Vetores de características, originados de diferentes conteúdos de um arquivo, são amplamente utilizados na classificação, e a abordagem *multi-view* tem mostrado melhorar a generalização e confiança do sistema.

Introdução

- **Objetivo geral:**
  - Apresentar um modelo de classificação de *malware Android* com *multi-view* e seleção de características com otimização multiobjetivo;
- **Objetivos Específicos:**
  - *Dataset* com 40 mil amostras de arquivos, 20 mil maliciosas e 20 mil benignas;
  - Módulo de análise estática para extração de características e criação dos vetores;
  - Módulo para seleção de características, com otimização multiobjetivo, e classificação *ensemble*.

Introdução

- **Contribuições:**
  - *Dataset* com 40 mil amostras de aplicativos *Android*, disponível publicamente, composto por permissões, api *calls* e o*pcodes*;
  - Modelo de detecção de *malware Android* implementado com *multi-view* e estratégia de otimização multiobjetivo. Nossa proposta melhorou a taxa verdadeiro positivo em uma média de 5,2, exigindo até 65% dos custos com processamento.

Tópicos

- Introdução
- **Background**
- Trabalhos Relacionados
- Proposta
- Resultados
- Conclusão
- Trabalhos Futuros
- Perguntas

Background

- Técnicas de aprendizado de máquina raramente são utilizadas em produção porque a classificação geralmente requer a análise de múltiplos indícios do conteúdo de um arquivo APK, enquanto a maioria das abordagens se concentra na análise de um único grupo de características, por exemplo, as permissões;
- A maioria dos modelos propostos, com múltiplos conjuntos de características, utiliza redes neurais profundas (DNN), que melhoram a precisão do sistema, embora impliquem desvantagens significativas em termos de requisitos de memória e processamento.

Tópicos

- Introdução
- Background
- **Trabalhos Relacionados**
- Proposta
- Resultados
- Conclusão
- Trabalhos Futuros
- Perguntas

Trabalhos Relacionados

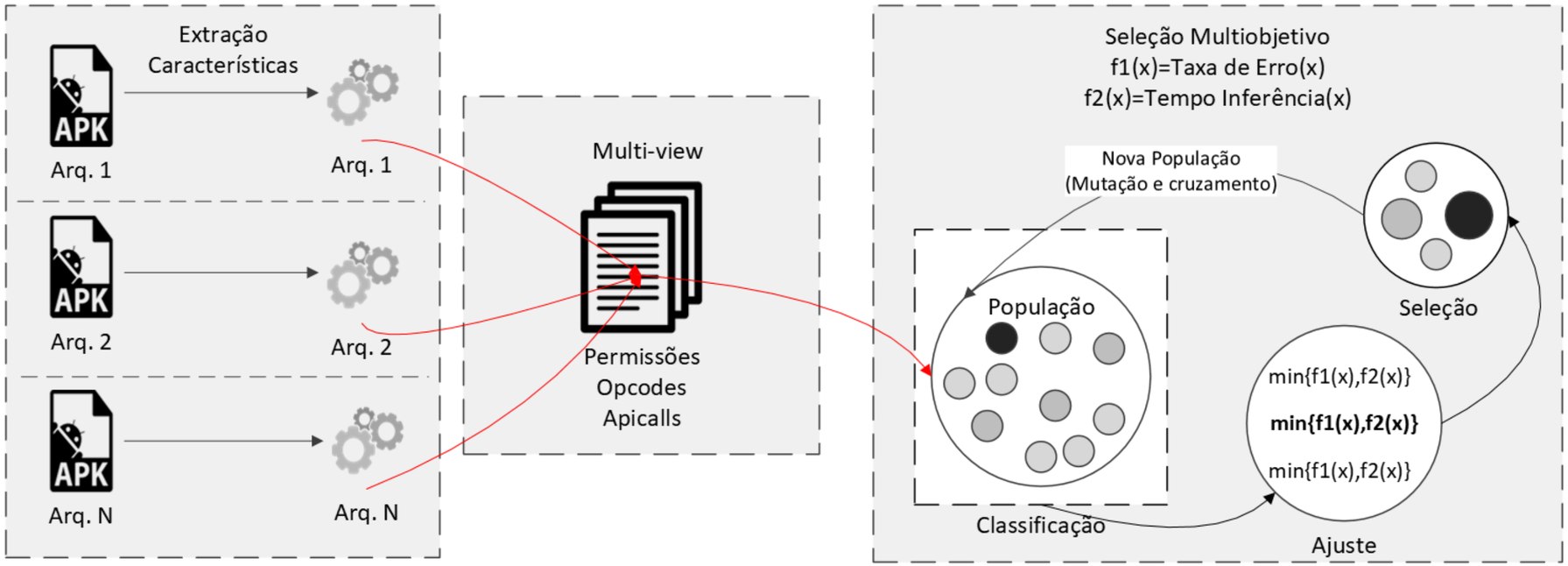
- Um grande número de trabalhos utilizam uma única *view* no processo de classificação:
  - Modelo de aprendizado de máquina baseado em permissões, com seleção de características, apresentou melhoras na precisão da classificação, porém utilizou um número limitado de amostras;
  - Modelo de DNN aprimorou a acurácia em comparação com estudos anteriores, mas não explorou *multi-view* para melhorar a generalização do modelo;

Tópicos

- Introdução
- Background
- Trabalhos Relacionados
- **Proposta**
- Resultados
- Conclusão
- Trabalhos Futuros
- Perguntas

Proposta

- Para superar o desafio da detecção de *malware Android* em um cenário *multi-view*, o esquema proposto é implementado através de uma abordagem de otimização multiobjetivo:



Proposta

- Extração de Características:
  - Utiliza-se o método de análise estática para extrair todas as características dos arquivos APK;
- Multi-view:
  - Nosso modelo utiliza três conjuntos de características (*views*) de um arquivo APK: permissão (manifest.xml), códigos de operação (dex) e chamadas de API (dex);
  - Cada *view* forma um vetor de características que será utilizado em modelos de classificação;

Proposta

- Seleção multiobjetivo:
  - O conjunto *multi-view* alimenta um pipeline de classificação:
    - aprendizado de máquina e seleção de características com otimização multiobjetivo;
  - Cada uma das *views* é classificada por um modelo (*Decision Tree* (DT) e *Random Forest* (RF)), produzindo um resultado de classificação para cada *view*, os quais são combinados por uma votação por maioria, produzindo o resultado final da classificação.

Proposta

- Seleção multiobjetivo:
  - A tarefa de otimização multiobjetivo visa encontrar um subconjunto de características para cada *view* de modo a minimizar o tempo de processamento e a taxa de erro.

Tópicos

- Introdução
- Background
- Trabalhos Relacionados
- Proposta
- **Resultados**
- Conclusão
- Trabalhos Futuros
- Perguntas

Resultados

- A avaliação do modelo orientou-se por três questões de pesquisa:
  - QP1: Qual é a precisão na detecção tradicional de *malware* Android utilizando aprendizado de máquina com uma única *view*?
  - QP2: De que maneira nossa abordagem de otimização multiobjetivo aprimora a acurácia da classificação?
  - QP3: Quais são as vantagens e desvantagens do nosso modelo em termos de custo de processamento, medido pelo tempo de inferência?

Resultados

- Para responder as questões de pesquisa, foi preciso inicialmente:
  - Criar um *dataset* para avaliar de forma confiável o desempenho do nosso modelo em um contexto de *multi-view*:
    - Amostras obtidas da plataforma AndroZoo;
    - 20 mil amostras maliciosas (selecionadas apenas quando 2 soluções no VirusTotal indicassem ser um *malware*);
    - 20 mil amostras benignas;

Resultados

- Extrair as características das amostras, tornando-as adequadas para o processo de classificação;
- Três views consideradas:
  - API calls: 63.460 características (.dex);
  - Opcode: 224 características (.dex);
  - Permissões: 19.083 características (manifest.xml);

Resultados

- O conjunto de dados inicial foi dividido em três subconjuntos:
  - Treino: 40%, utilizado para construir os classificadores;
  - Teste: 30%, utilizado no processo de otimização multiobjetivo;
  - Validação: 30%, utilizado para avaliar o sistema final.
- Construção dos modelos de classificação, parâmetros:
  - DT: critério de gini, sem limite de profundidade;
  - RF: 100 árvores de decisão, cada uma com os mesmos parâmetros do classificador DT.

Resultados

- Normalização dos *datasets* aplicando o método min-max;
- Dimensionalização dos *datasets* aplicando PCA com 100 componentes;
- Construção do módulo de seleção de características com multiobjetivo:
  - Algoritmo NSGA-II, utilizando a API pymoo;
  - População de tamanho 100; 100 gerações; taxa de cruzamento 0,3; probabilidade de mutação 0,1;

Resultados

- O problema do NSGA-II foi parametrizado com 300 variáveis e 2 objetivos; Cada variável pode assumir valor entre 0 e 1;
- Cada 100 variáveis representa um vetor de características de uma *view*;
- Se o valor da variável for menor que 0,5 (50%) a característica é excluída do vetor;

- A avaliação dos classificadores foi baseada em suas taxas de *True Positive* (TP) e *True Negative* (TN):
  - TP: amostras de *malware* corretamente classificadas;
  - TN: amostras de *goodware* corretamente classificadas;

Resultados

- QP1: Qual é a precisão na detecção tradicional de *malware Android* utilizando aprendizado de máquina com uma única *view*?
  - RF:
    - Api calls: 83,17% TP;
    - Opcode: 84,32% TP;
    - Permissões: 78,45% TP;
- Resultado QP1: não oferece o nível necessário de confiança para uma implementação em ambiente de produção.

Resultados

- QP2: De que maneira nossa abordagem de otimização multiobjetivo aprimora a acurácia da classificação?
  - RF:
    - *Multi-view*: 87,22% TP;
- Resultado QP2: nossa abordagem demonstrou uma melhoria na precisão da classificação; Em média, nossa metodologia elevou os índices TP em 5,2 pontos.

Resultados

- Visão Geral dos resultados:

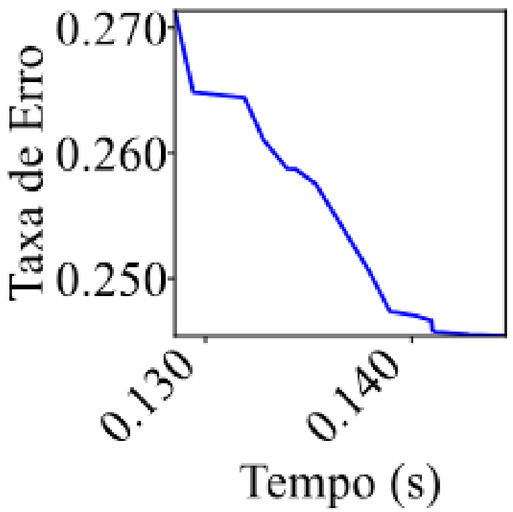
View	Classificador	Acurácia (%)		
		TP	***TN***	***F1***
API Calls	DT	70.91	71.02	70.95
	RF	83.17	77.18	80.84
Opcode	DT	70.34	70.91	70.45
	RF	84.32	76.57	81.16
Permissões	DT	68.43	68.82	68.49
	RF	78.45	75.14	77.41
Nossos Resultados	DT	75.86	75.35	75.74
	RF	87.22	78.18	83.24

Resultados

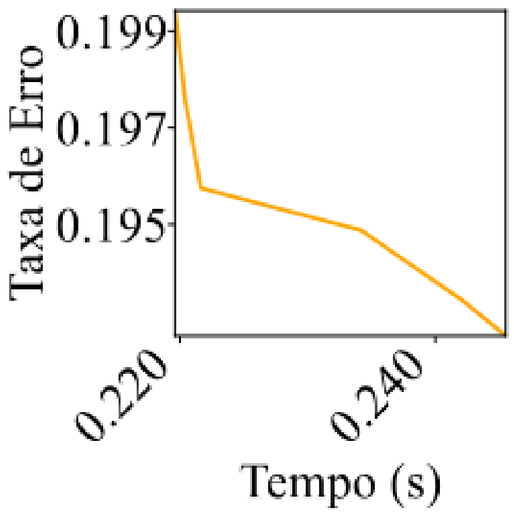
- QP3: Quais são as vantagens e desvantagens do nosso modelo em termos de custo de processamento, medido pelo tempo de inferência?
  - Exigiu apenas 65%, 78% e 66% dos custos de processamento correspondentes ao uso de todas as características com os classificadores DT e RF;
- Resultado QP3: o modelo pode reduzir os custos de processamento de inferência associados.

Resultados

- Curva de Pareto do modelo proposto. Observa-se uma correlação direta entre custos de processamento de inferência e a taxa de erro do sistema:



(a) DT *Multi-view*



(b) RF *Multi-view*

Tópicos

- Introdução
- Background
- Trabalhos Relacionados
- Proposta
- Resultados
- **Conclusão**
- Trabalhos Futuros
- Perguntas

Conclusão

- A seleção de um subconjunto de características de cada *view* maximiza a precisão da classificação quando combinada através de um conjunto de classificadores;
- As experiências conduzidas em um novo *dataset* demonstram a viabilidade da nossa proposta, resultando em uma melhora significativa na precisão e uma redução nos custos computacionais de inferência.

Tópicos

- Introdução
- Background
- Trabalhos Relacionados
- Proposta
- Resultados
- Conclusão
- **Trabalhos Futuros**
- Perguntas

Trabalhos Futuros

- Mestrado:
  - Inicio do mestrado como aluno ouvinte;
  - Continuação da pesquisa com foco em detecção de *malwares Android* baseado em análise multimodal:
    - estudo da literatura;
    - delimitação do problema;
    - em linhas gerais: características provenientes de diferentes origens (estática, dinâmica, metadados), estratégias para lidar com a ausência de características, modelos de deep learning.

Tópicos

- Introdução
- Background
- Trabalhos Relacionados
- Proposta
- Resultados
- Conclusão
- Trabalhos Futuros
- **Perguntas**

Obrigado!

Philipe Fransozi
Jhonatan Geremias
Eduardo K. Viegas
Altair Santin

{philipe.hfransozi, jgeremias, eduardo.viegas, santin}@ppgia.pucpr.br

Perguntas!

Patrocinadores do SBSeg 2024!

nic.br

cgi.br

Google

Tempest

CAPES

SiDi

FAPESP

zscaler™

BugHunt

CNPq

C.E.S.A.R

FACULDADE IBPTECH