



Performance Improvement of Repackaged Android Application Detection Techniques

Presenter : Mojtaba Moazen

Supervisor : Dr. Morteza Amini

Repackaging

- Downloading The Original Application
- Accessing source code by Reverse Engineering
- Adding Malicious Code or Other Types of Unauthorized Functionality
- Repack to the new APK file and publish it



Repackaging aims

User

Abusing users

Malwares execution

User's security

Android Developer

Economic threats

change payments API

Adware

Obfuscation



- **definition**
- **Changing Original Application**
 - Resources level (User Interfaces)
 - Sources Level (Logic)

```
function myFunc(str) {  
    document.write(str);  
}  
var myStr = "My Code";  
myFunc(myStr);
```

```
function msfrt23kjgty(zs12mnjy) {  
    document.write(zs12mnjy);  
}  
var nbuqmazsuikh = "My Code";  
msfrt23kjgty(nbuqmazsuikh);
```

Type of Obfuscation

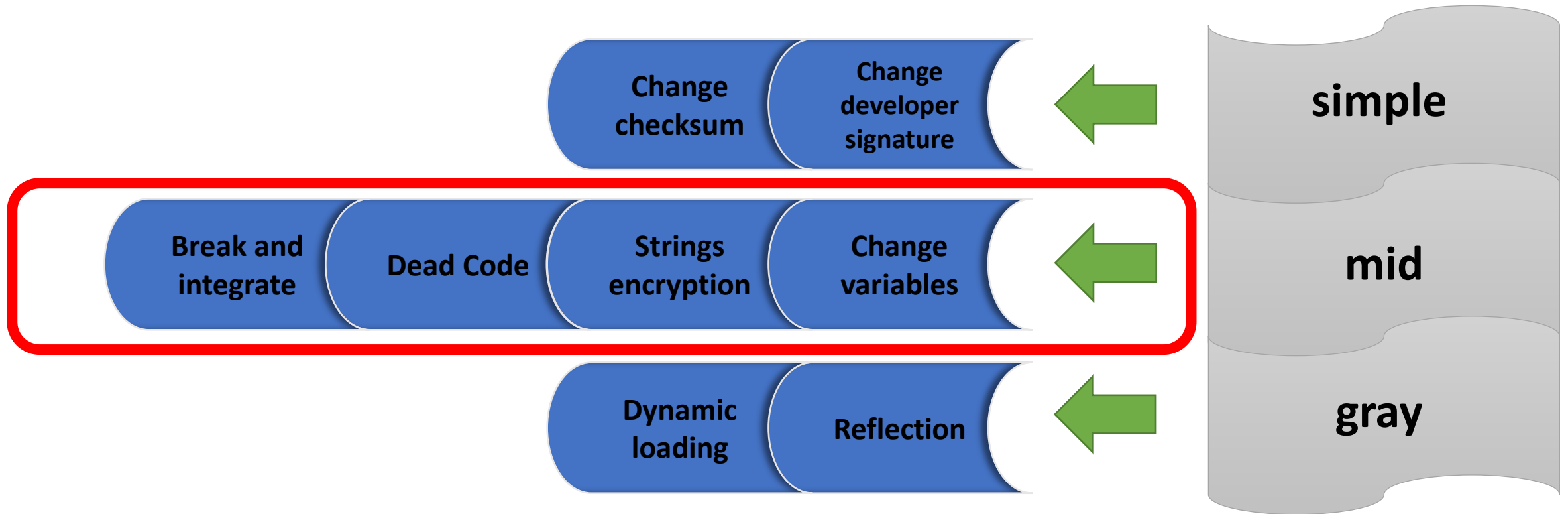
- **Developers**



- **Attackers**

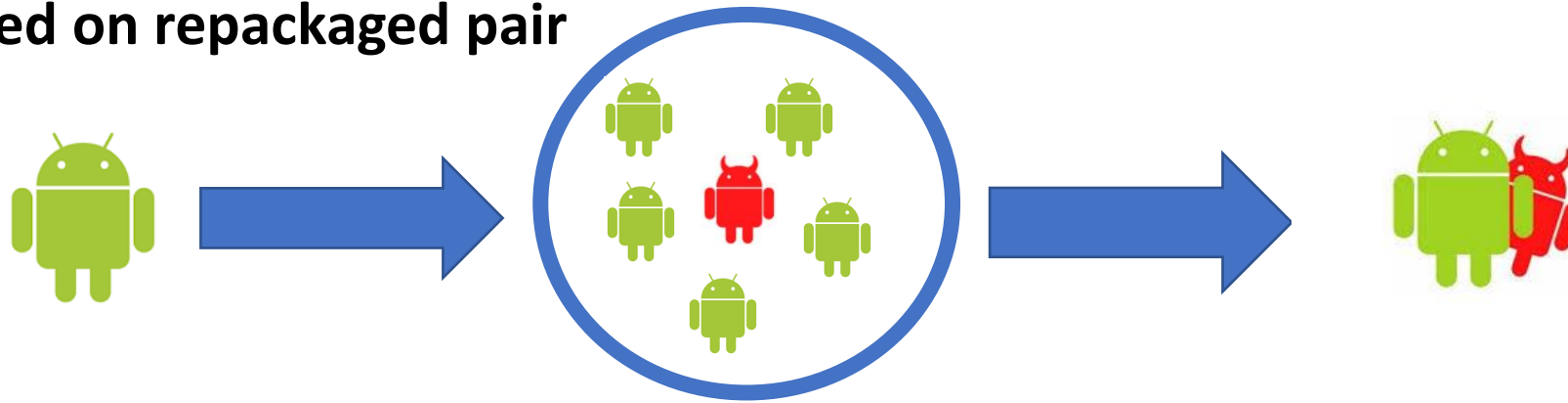


Levels of obfuscation

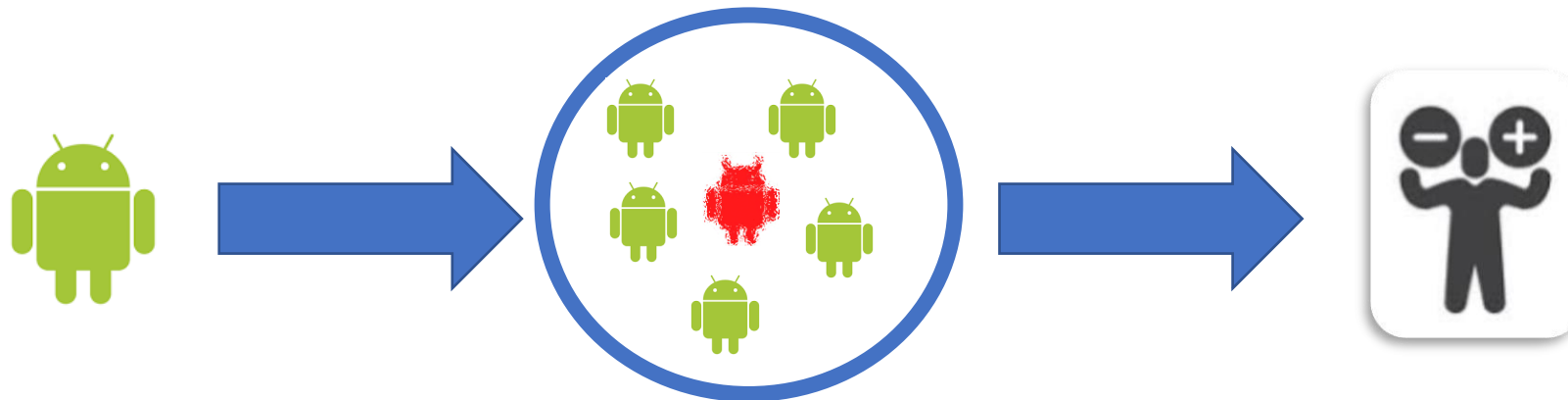


Repackaging Detection Definition

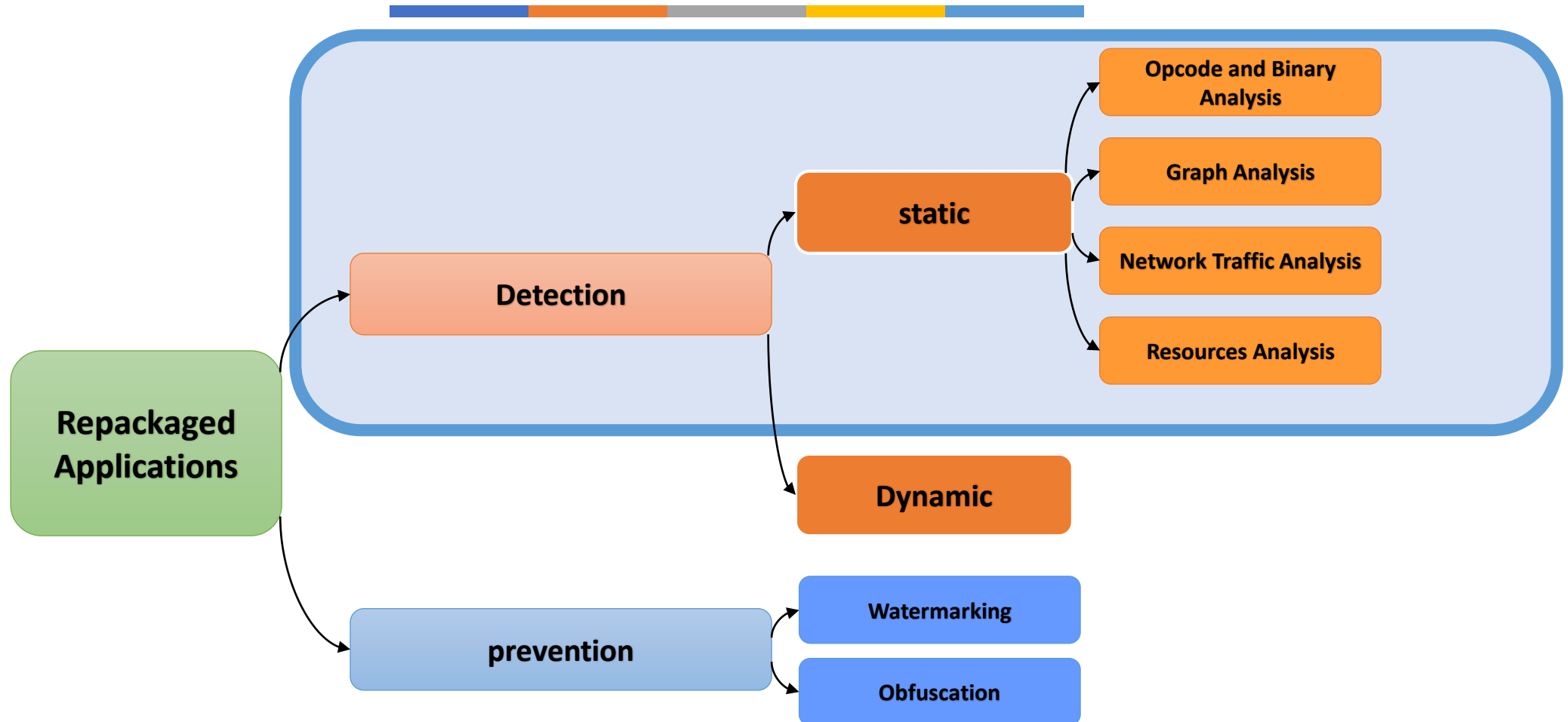
- **Based on repackaged pair**



- **Based on Decision**



Related Works



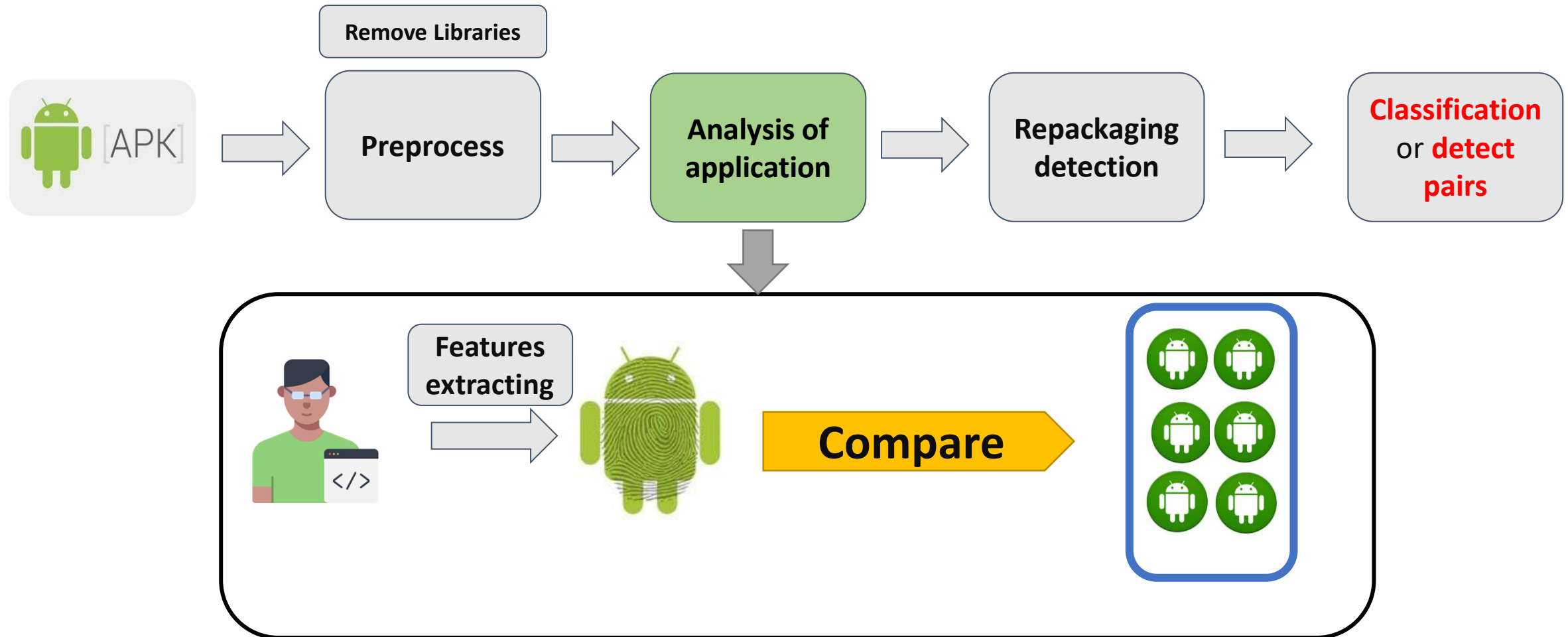
Problems with existing approaches



- **Low accuracy and recall**
- **Limited definition about obfuscation's target**
- **Limited definition about detection repackaged apps**
- **Hight accuracy , high execution time**

We've been convinced to propose an accurate method with less execution time

Static Analysis



Signature Generation

Method Signature

$\langle \textit{Modifier}, \textit{RetType}, \textit{InputType}, \textit{JLibMethodCallee}, \textit{NonStaticAppMethodcallee}, \textit{ApiCallSootSignature} \rangle$

$\langle \textit{DeclaringClass}, \textit{RetType}, \textit{MethodName} \rangle$

```
graph TD; A["< Modifier, RetType, InputType, JLibMethodCallee, NonStaticAppMethodcallee, ApiCallSootSignature >"] --> B["< DeclaringClass, RetType, MethodName >"]
```

Method Declaration

```
public Object push(Object item)
{
    items.addElement(item);
    return item;
}
```

Method Body

```
graph LR; A[Method Declaration] --> B["public Object push(Object item)"]; B --> C["{ items.addElement(item); return item; }"]; C --> D[Method Body]
```

Signature Generation



Class Signature

$\langle \textit{ClassCoreSig}, \textit{InnerClassesSig}, \textit{InheritedClassesSig}, \textit{ImpInterfacesSig}$
 $, \textit{ClassLevel}, \textit{InnerOuterClassName}, \textit{ClassLen}, \textit{NumofInnerClass} \rangle$

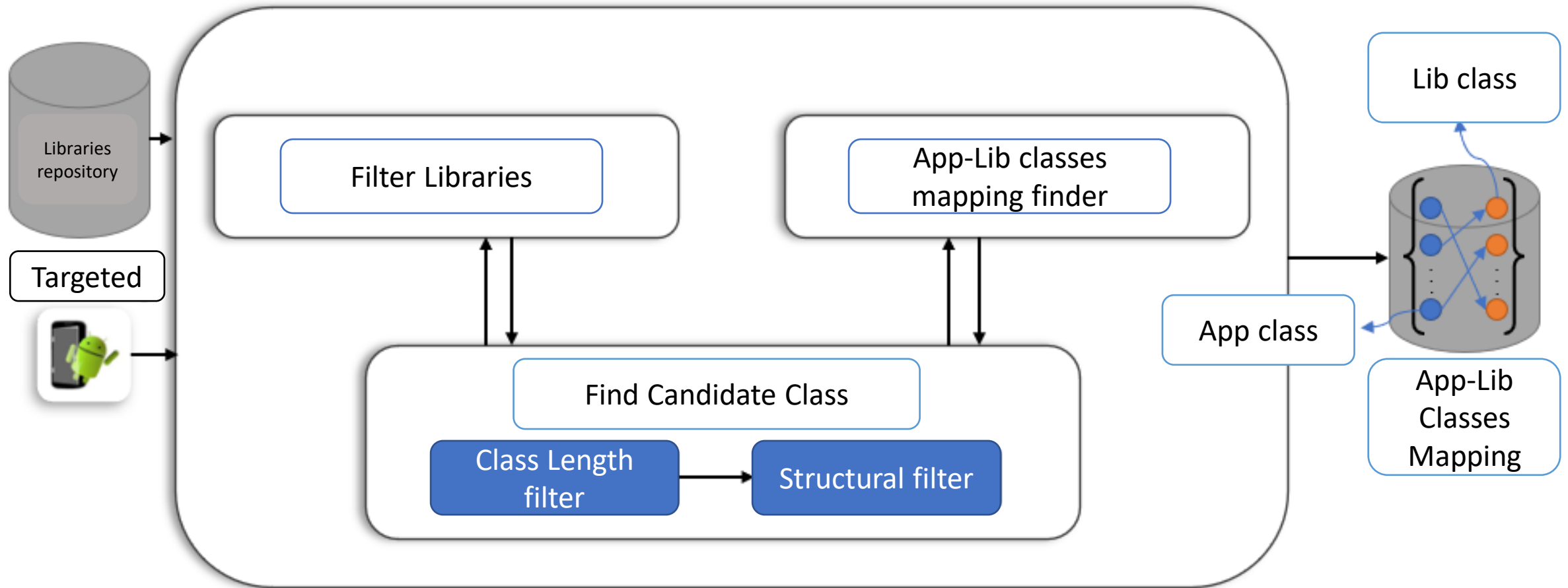
ClassCoreSig

$\langle \textit{MethodSig}_1, \textit{MethodSig}_2, \textit{MethodSig}_3, \dots, \textit{MethodSig}_n \rangle$

Application's Signature

$\langle \textit{Classsig}_1, \textit{ClassSig}_2, \textit{ClassSig}_3, \dots, \textit{ClassSig}_n \rangle$

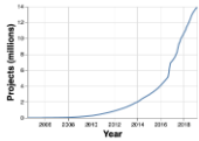
Application's libraries detection Component



Libraries repository

- We made a dataset of 877 libraries with their dependency from maven


MVN REPOSITORY[Categories](#) | [Popular](#) | [Contact Us](#)

Indexed Artifacts (32.7M)


Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- Core Utilities
- JVM Languages
- Mocking
- Language Runtime
- Web Assets
- Annotation Libraries
- Logging Bridges
- HTTP Clients
- Dependency Injection
- XML Processing
- Web Frameworks
- I/O Utilities
- Defect Detection Metadata


What's New in Maven

**AWS Java SDK For AWS Secrets Manager**
[com.amazonaws » aws-java-sdk-secretsmanager » 1.12.425](#)
The AWS Java SDK for AWS Secrets Manager module holds the client classes that are used for communicating with AWS Secrets Manager Service.
Last Release on Mar 11, 2023


50 usages
Apache

**CDK8s**
[org.cdk8s » cdk8s » 1.10.26](#)
This is the core library of Cloud Development Kit (CDK) for Kubernetes (cdk8s). cdk8s apps synthesize into standard Kubernetes manifests which can be applied to any Kubernetes cluster.
Last Release on Mar 11, 2023


18 usages
Apache

**CDK8s**
[org.cdk8s » cdk8s » 2.7.27](#)
This is the core library of Cloud Development Kit (CDK) for Kubernetes (cdk8s). cdk8s apps synthesize into standard Kubernetes manifests which can be applied to any Kubernetes cluster.
Last Release on Mar 11, 2023

18 usages
Apache

**AWS Java SDK For Amazon Cognito Identity Provider Service**
[com.amazonaws » aws-java-sdk-cognitoidp » 1.12.425](#)
The AWS Java SDK for Amazon Cognito Identity Provider Service module holds the client classes that are used for communicating with Amazon Cognito Identity Provider Service.
Last Release on Mar 11, 2023

15 usages
Apache

**Segment Kotlin Analytics**
[com.segment.analytics.kotlin » android » 1.10.3](#)
The hassle-free way to add analytics to your Kotlin app.
Last Release on Mar 11, 2023

14 usages
MIT

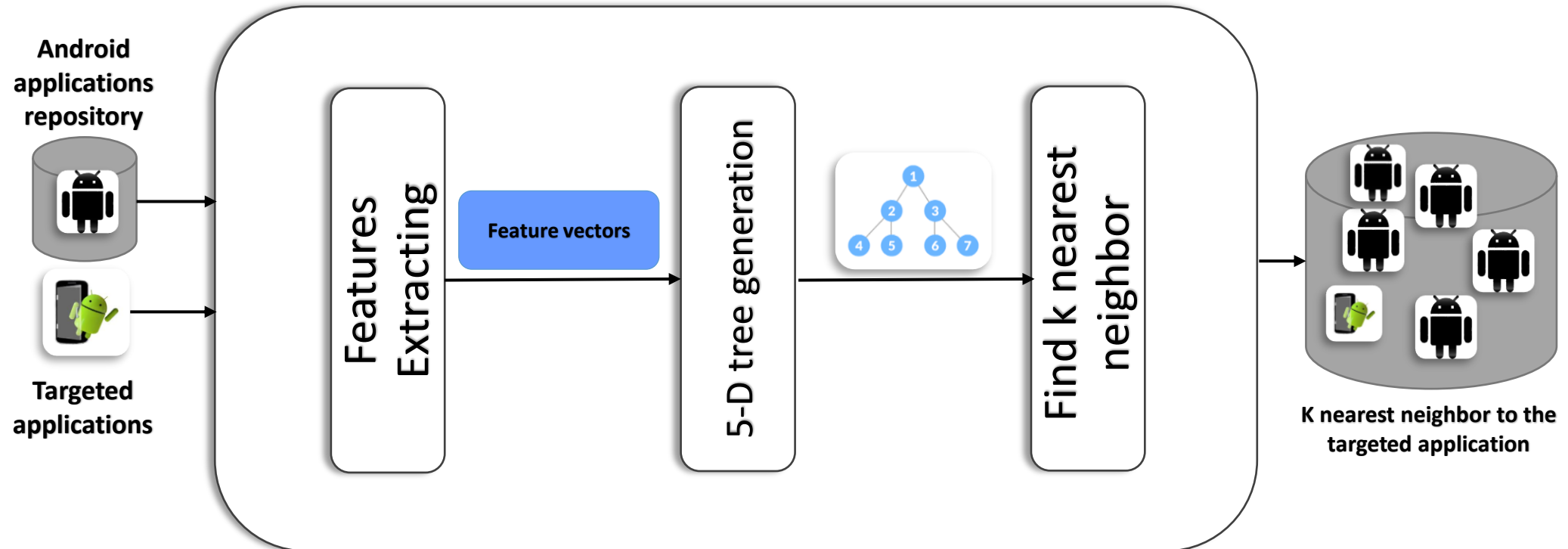
Indexed Repositories (1888)

- Central
- Atlassian
- Sonatype
- Hortonworks
- Spring Plugins
- Spring Lib M
- JCenter
- JBossEA
- Atlassian Public
- BeDataDriven

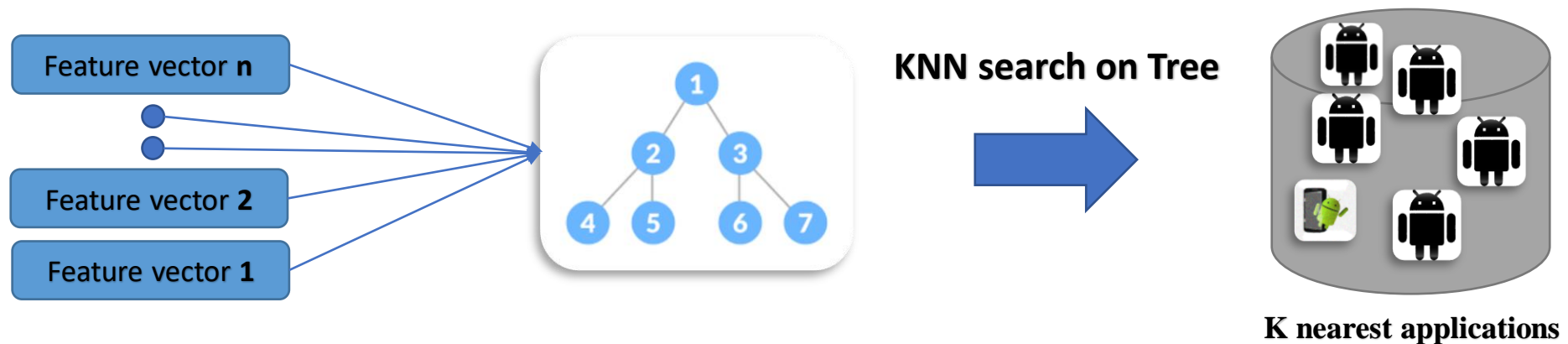
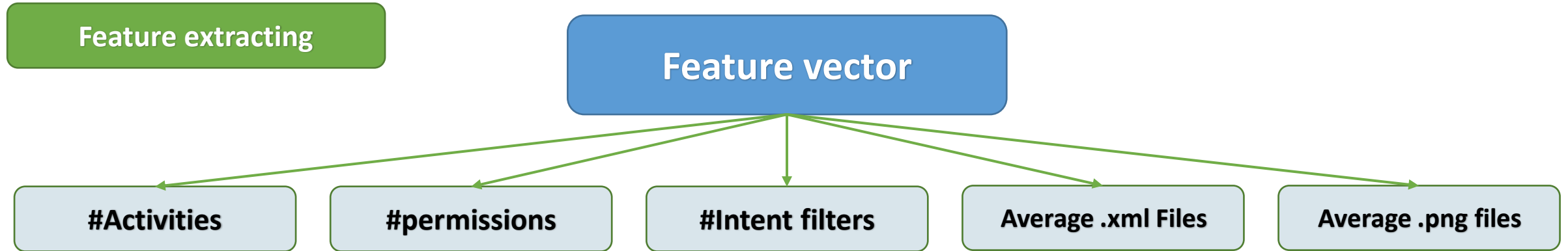
Popular Tags

- aar amazon android apache api
- application arm assets atlassian aws
- build build-system client clojure

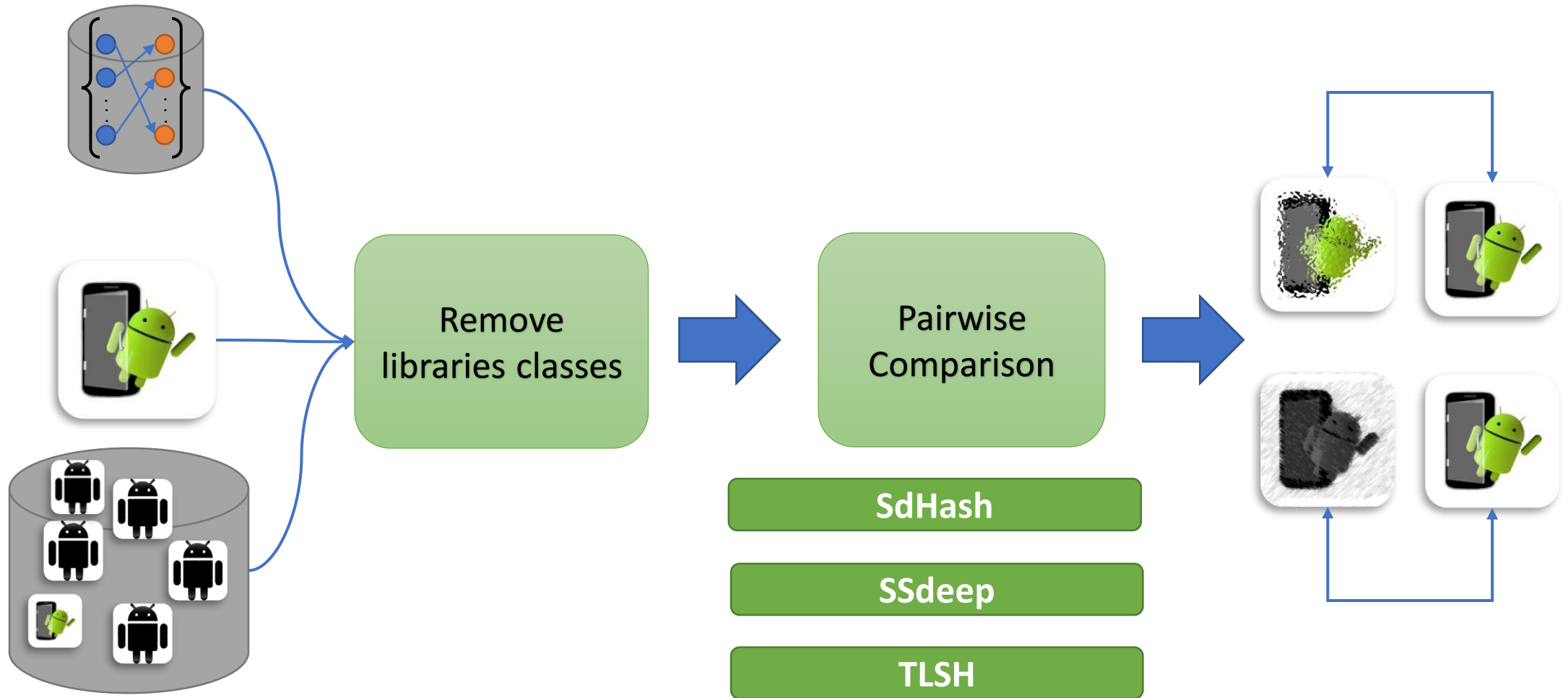
KNN Classifier



KNN Features

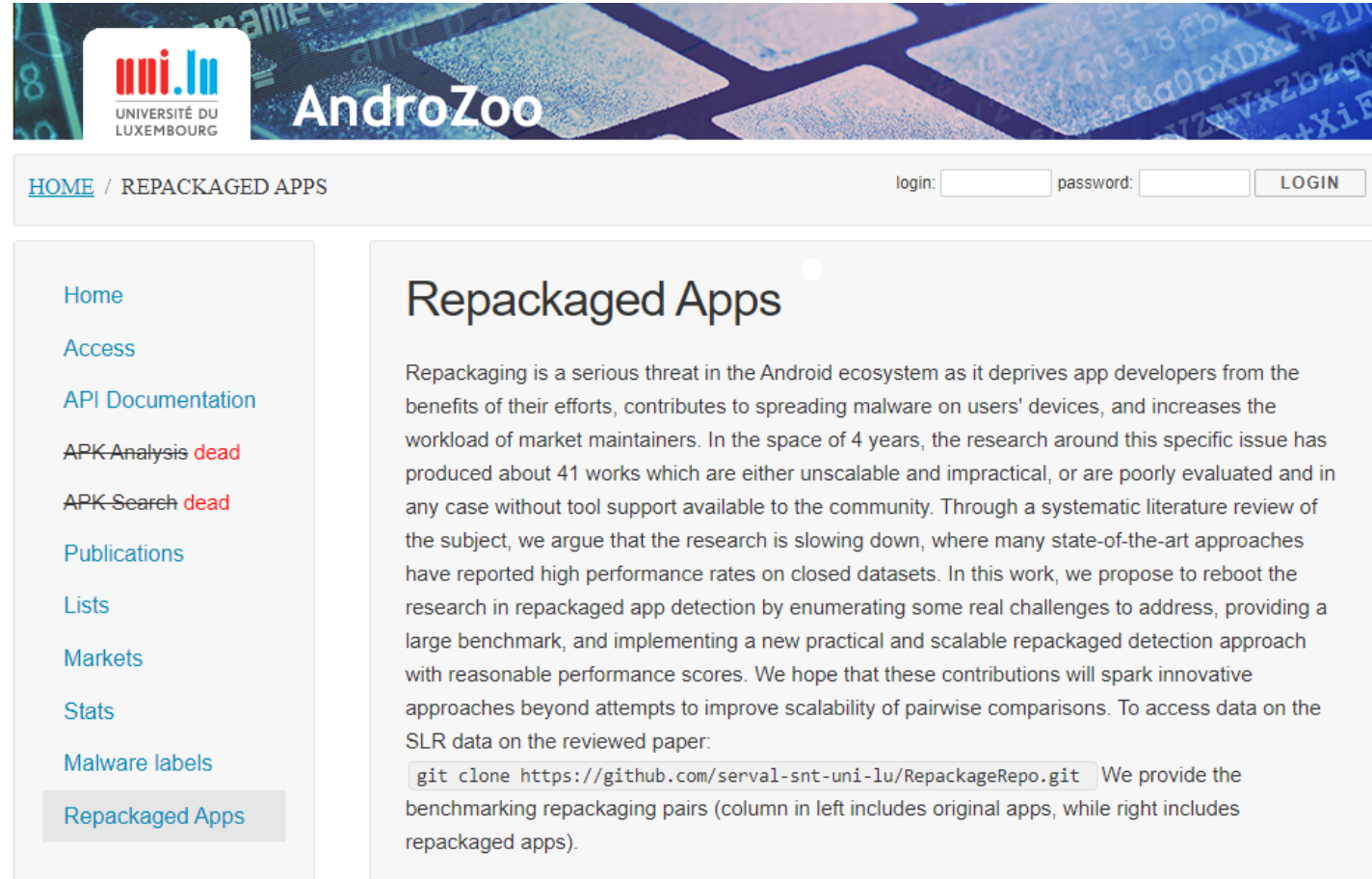


Pairwise Comparison




Dataset

- 1181 android applications
- 1196 pairs
- 400 non-repackaged pair
- 796 repackaged pairs
- 877 java and android libraries



Execution time without classification

- 
- 1 • Generate Signature
 - 2 • Find Library's classes
 - 3 • Hashing Signature
 - 4 • Compare Hashes

	Ssdeep	Sdhash	TLSH
Generate Signature	5	5	5
Find Libraries Classes	18	164	71
Hashing (signature without libraris)	0.23	0.71	0.81
Compare Hashes	0.1	1	0.26
Average Total Execution time(second)	24	171	78

Execution time without classification



	Our approach	Torki's approach
Average Total Execution time	24	126

5.25x improvement of execution time

Precision and Recall



Our work	Ssdeep	Sdhash	TLSH
Precision	97%	96%	97%
Recall	96%	94%	97%

Compare torki's method to our	Our Work	Torki's Method
Precision	97%	98%
Recall	96%	96%

1% = 5x better execution time

Pairwise Comparison Using KNN Classifier



- Considering 250 nearest application to each targeted application
- 97% of repackaged pairs placed in maximum 250 nearest applications
- 5x reducing comparison space(250 first application instead of 1180)

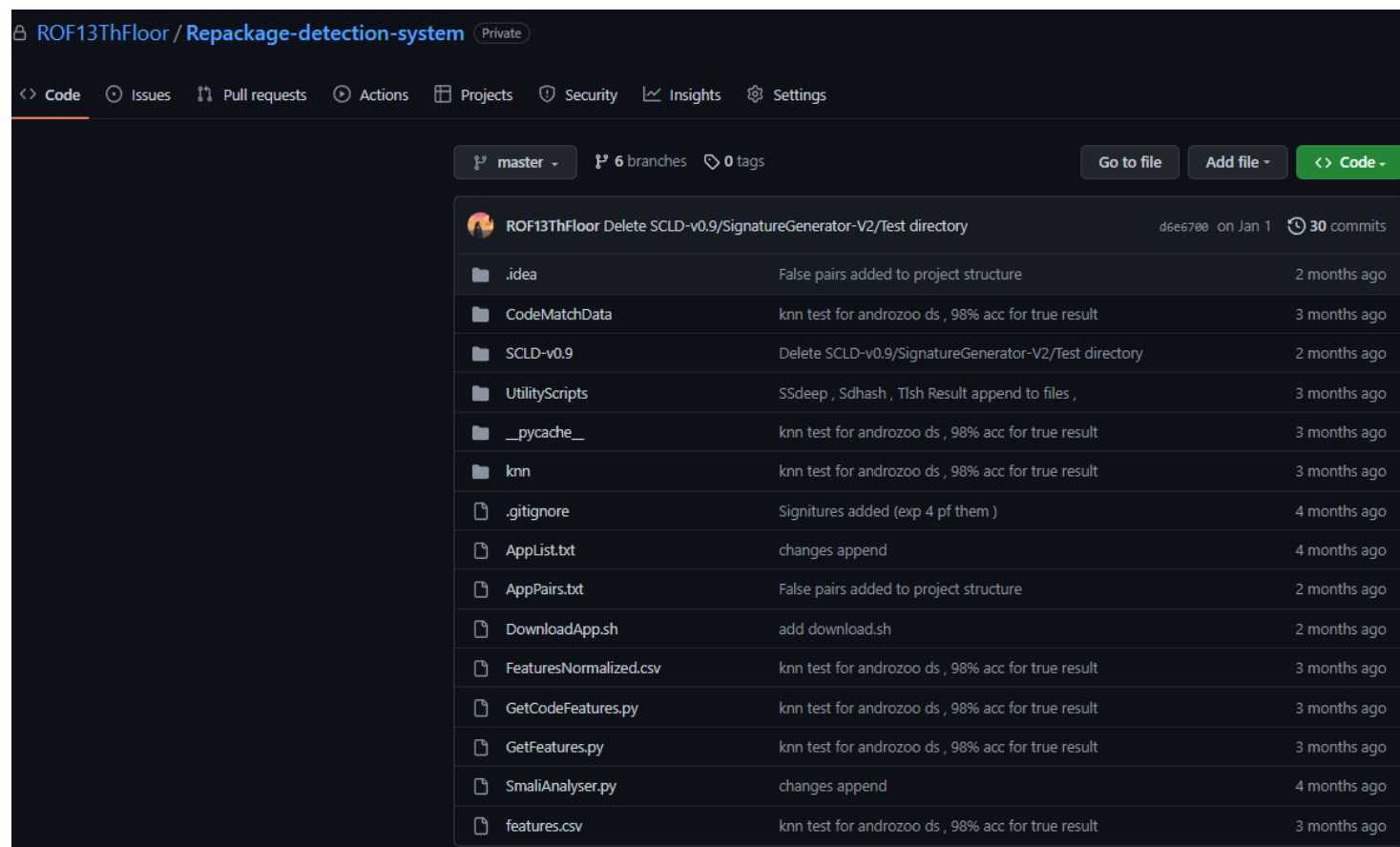
Discussion

- Proposed a new method based on coarse grained classification method
- 5x Acceleration of previous work by proposed a new signature of application
- 5x reduction of comparison space by integrating KNN with Pairwise Comparison
- Made a new dataset of 877 applications which can be use in related ongoing research



Future Work

- Weakness of dataset
- Commercial Applications are more complicated
- Proposed approach depends on call graph
- Feature engineering methods to extract better features





Email : mojtaba.moazen.a@gmail.com



Github : [rof13thfloor](#)