

Malware Detection in Android Applications with Machine Learning Techniques

Thesis presentation

Catarina Rodrigues Palma

Mestrado em Engenharia Informática e de Computadores
Instituto Superior de Engenharia de Lisboa

Júri

Orientador: Professor Doutor Artur Jorge Ferreira

Presidente: Professor Doutor Tiago Miguel Braga da Silva Dias

Vogais: Professor Doutor Rui Manuel Feliciano de Jesus

Professor Doutor Artur Jorge Ferreira



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

December 14th, 2023

Summary

- Context and Motivation
- Goals
- Proposed Approach
- Datasets
- Experimental Results
- Conclusions and Future Work
- Contributions

Context and Motivation

- 70% of mobile phones use the Android operating system (OS)
- In Q3 2022, Google Play Store hosted around 3.5 million applications (apps)
- These apps deal with a great amount of user-sensitive data
- Thus, they are a prized target for malicious software (malware) developers
- In 2020, 5.7 million Android malware packages were detected, tripling 2019's 2.1 million



Context and Motivation (2)

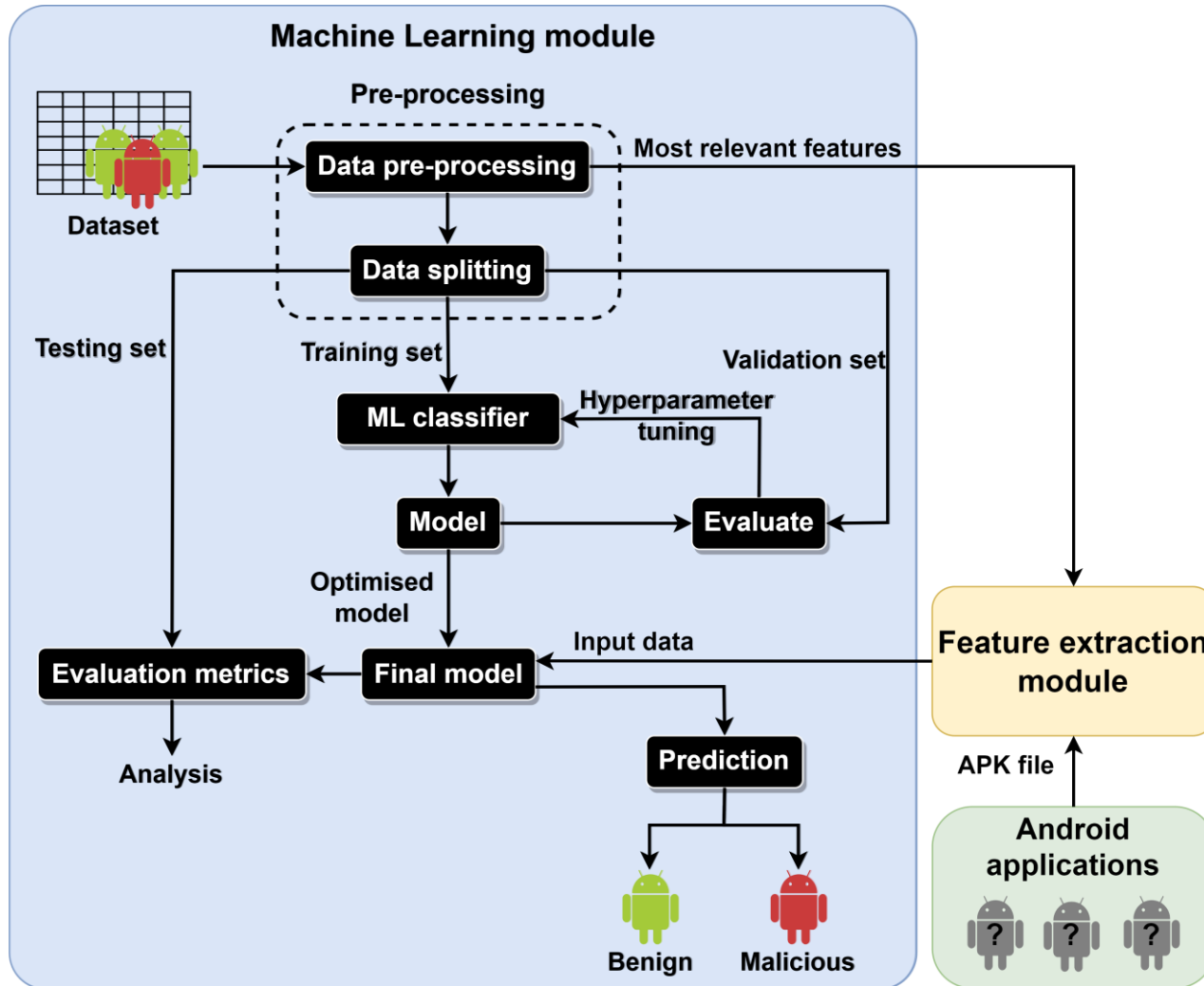
- Existing security measures to mitigate malware are to some extent successful
- However, malware keeps growing in both sophistication and diffusion, sometimes easily bypassing security measures
- Machine Learning (ML) approaches are known to be efficient and versatile
- Thus, we explore the use of ML techniques to detect malware in Android apps

Goals

- Identify the most decisive features for Android malware classification
- Recognise the ML classifiers that provide the most satisfying results in detecting malware in Android applications
- Assess the impact of different data pre-processing techniques applied to this problem's datasets
- Develop a prototype that resorts to ML techniques to detect malware in Android applications

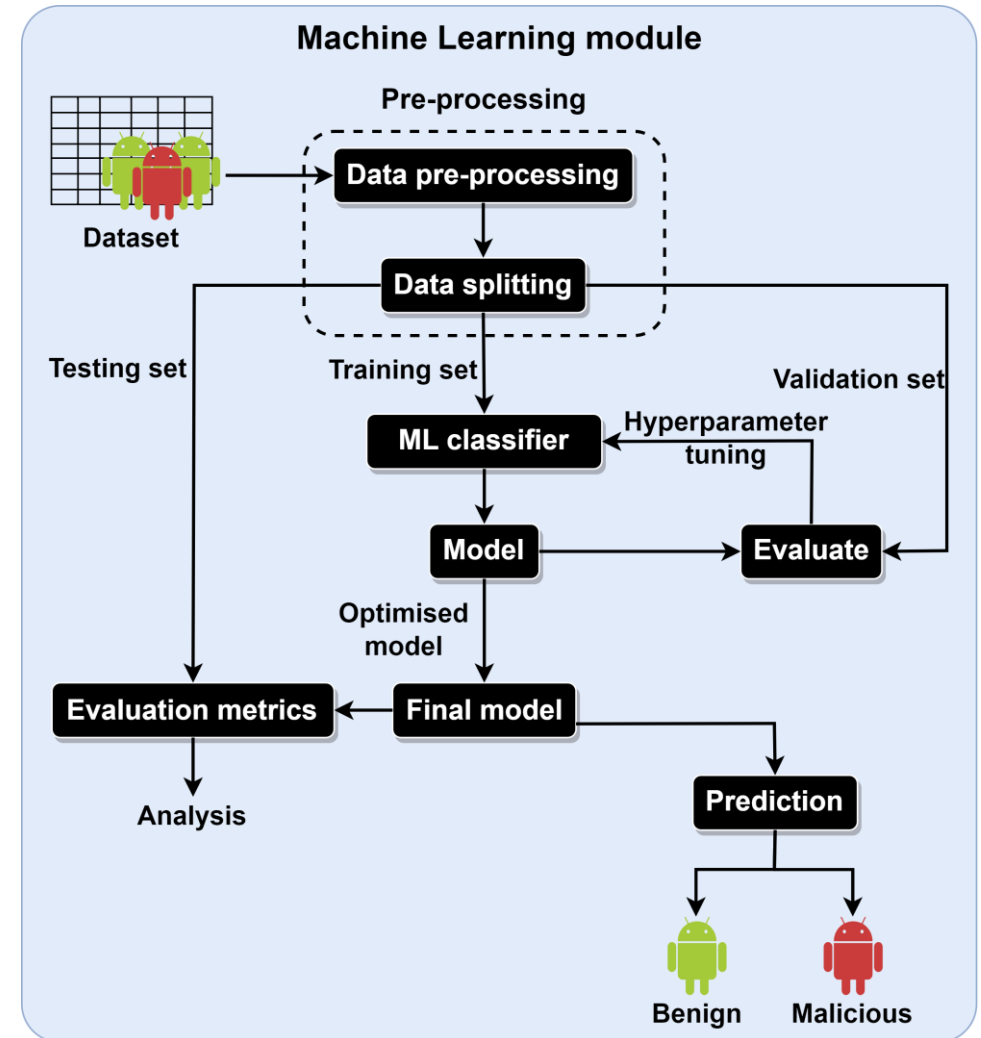


Proposed Approach



- Formulated as a binary classification problem
- The proposed approach can be divided into two major parts:
 - Machine Learning module
 - Feature extraction module and Android applications





Machine Learning module



Datasets – Data Acquisition

- Features can be obtained via different types of analysis:
 - **Static analysis** - Analysis in a non-runtime environment
 - **Dynamic analysis** - Requires the Android application to be running
 - **Hybrid analysis** - Combination of the two previous analysis
- Static analysis is the most common approach
- A static analysis approach is followed, thus, dynamic features were removed from the datasets

Dataset analysis & Data processing

	Drebin	CICAndMal2017	Android Malware (AM)	Android Malware static feature (AMSF)
instances (n)	15036	29999	11476	5019
features (d)	215	110	182	966
Release year	2014	2018	2016	-
Categorical features	1	5	12	0
Missing values	0	204	19888	0
Class label majority	 63.02%	 66.67%	 70.22%	 50.03%

- The following experiments were performed after:
 - Converting categorical features to numerical via label encoding
 - Removal of the instances containing missing values

Evaluation Metrics

- **Confusion Matrix**

- True Positive (TP) – Malicious app as malicious
- False Positive (FP) – Benign app as malicious
- True Negative (TN) – Benign app as benign
- False Negative (FN) – Malicious app as benign

- **Accuracy**

- correct predictions rate

- **Recall**

- true positive rate

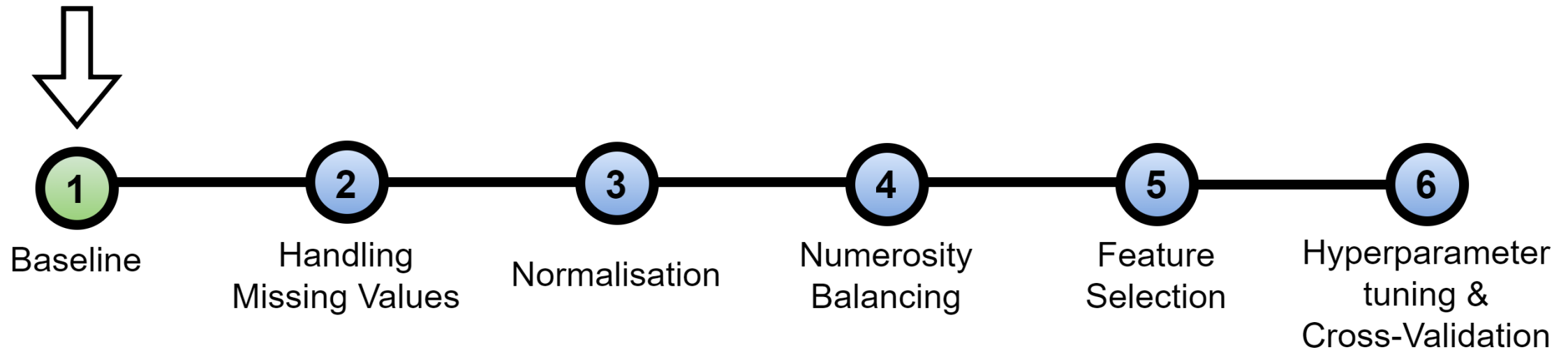
- **Precision**

- positive predictive value

- **F1-Score**

- **AUC-ROC** (Area Under the Curve-Receiver Operating Characteristic)

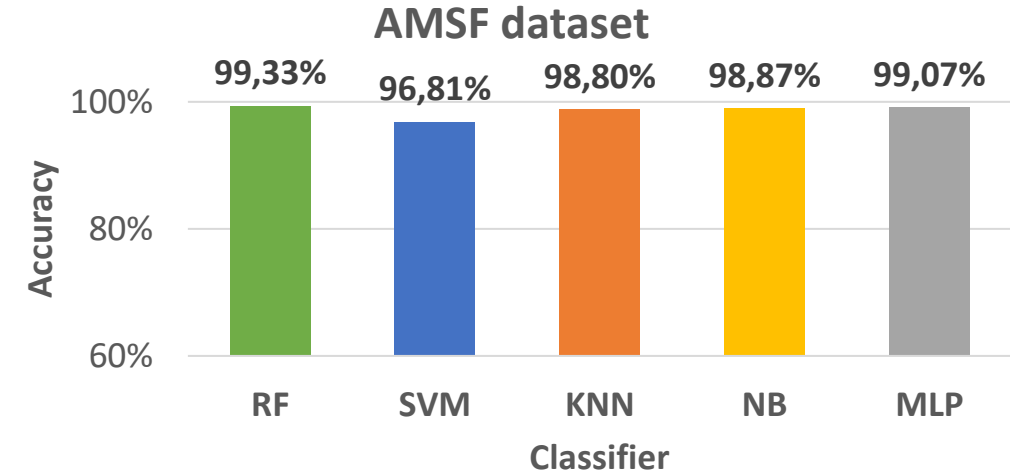
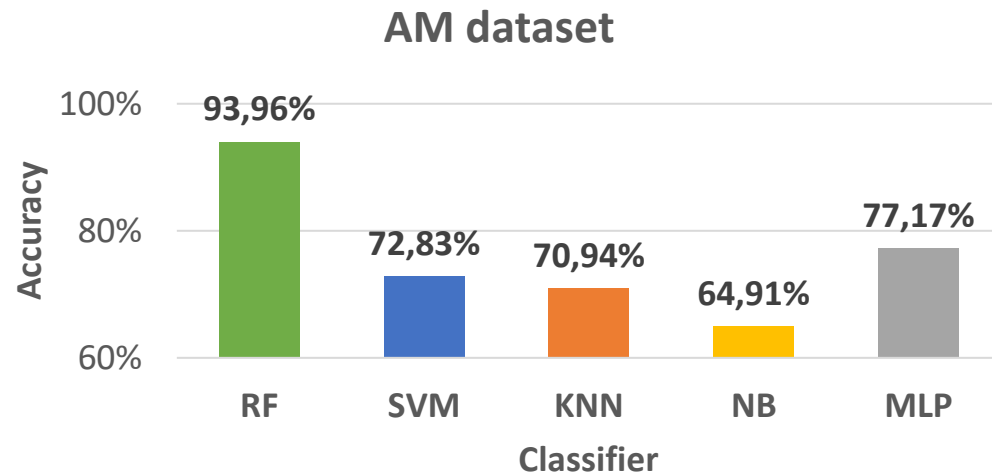
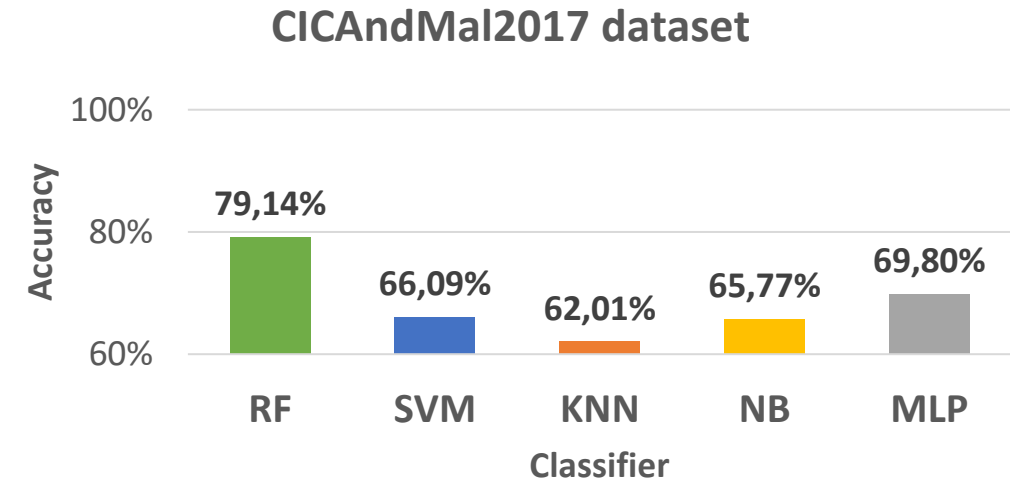
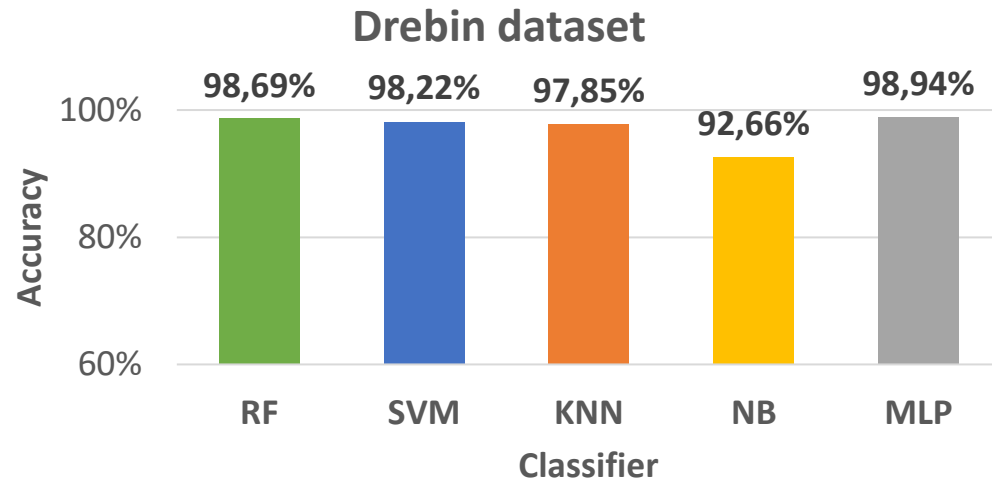
Experimental Results: Baseline



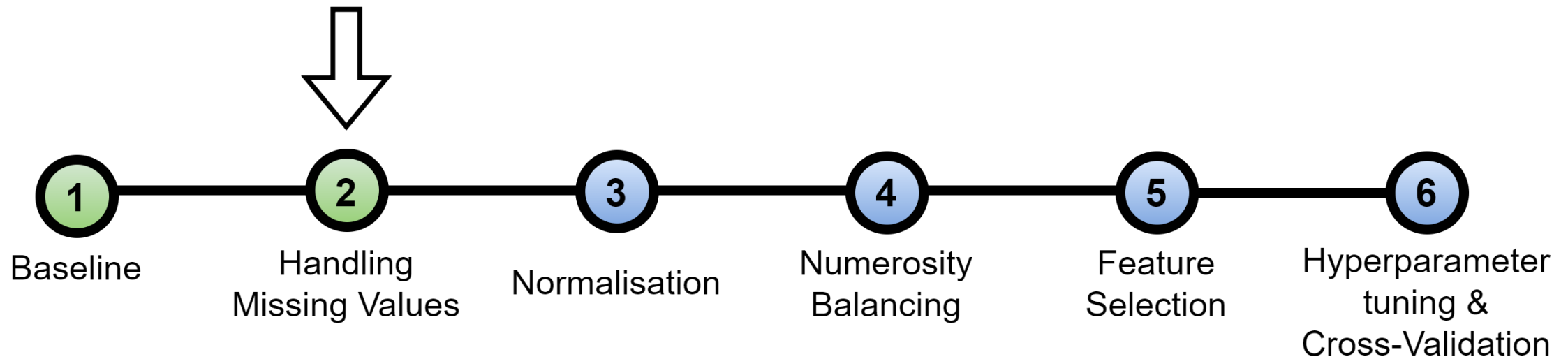
Experimental Results: Baseline

- Stratified random split of 70-30 for train-test was applied
- Experiments with different classifiers:
 - Random Forest (RF)
 - Support Vector Machines (SVM)
 - K-Nearest Neighbours (KNN)
 - Naive Bayes (NB)
 - Multi-layer Perceptron (MLP)

Experimental Results: Baseline (2)

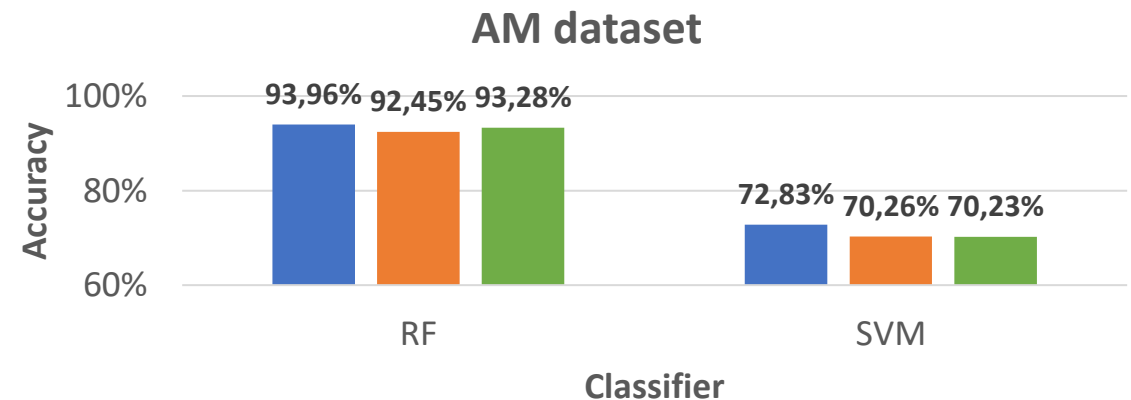
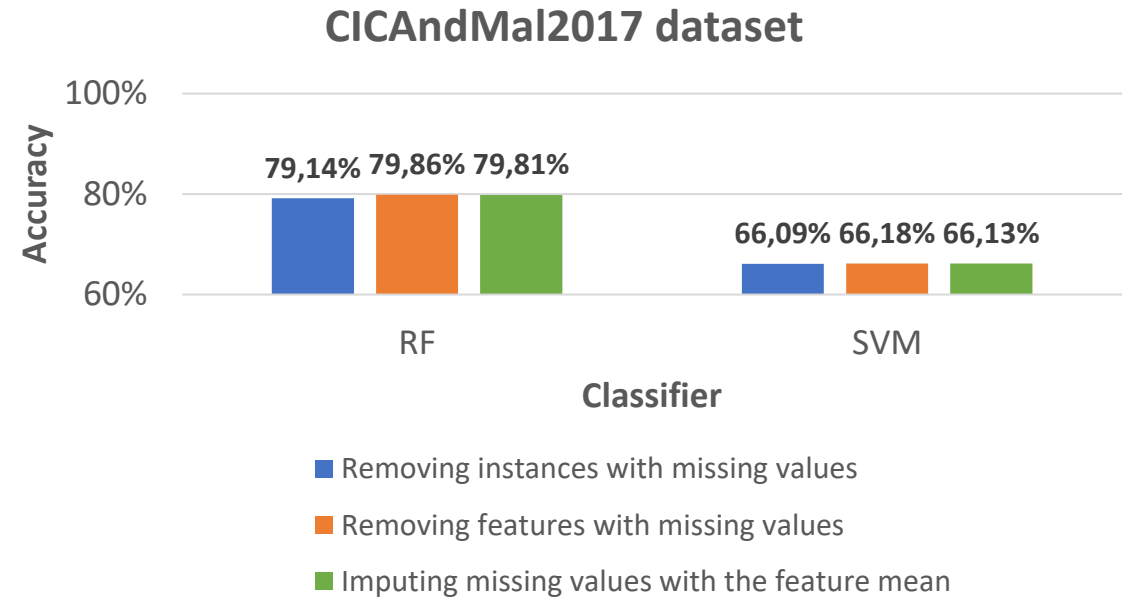


Experimental Results: Handling Missing Values

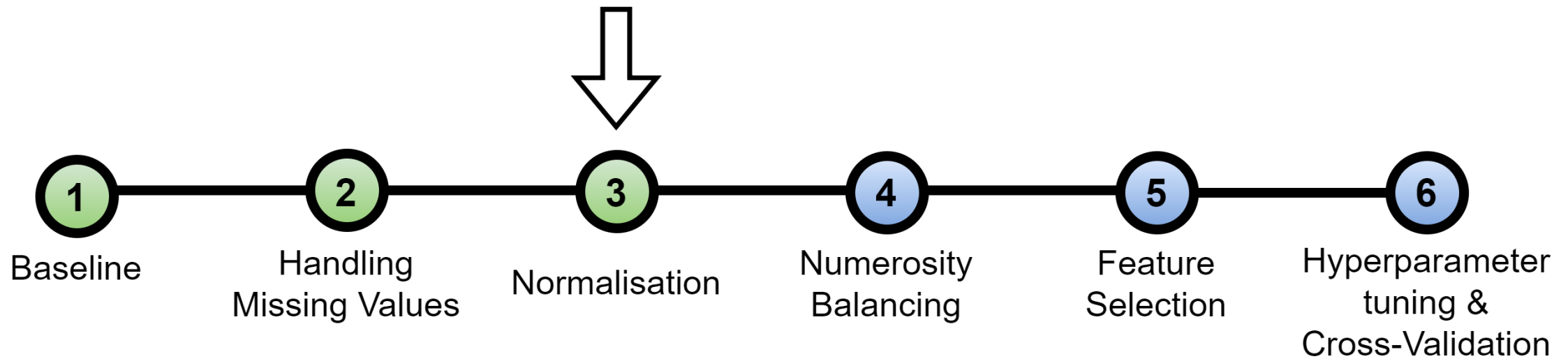


Experimental Results: Handling Missing Values

- Experiments with different methods to deal with missing values
- The results do not differ significantly between the tested methods
- May be an indicator of the presence of redundant and/or irrelevant features



Experimental Results: Normalisation

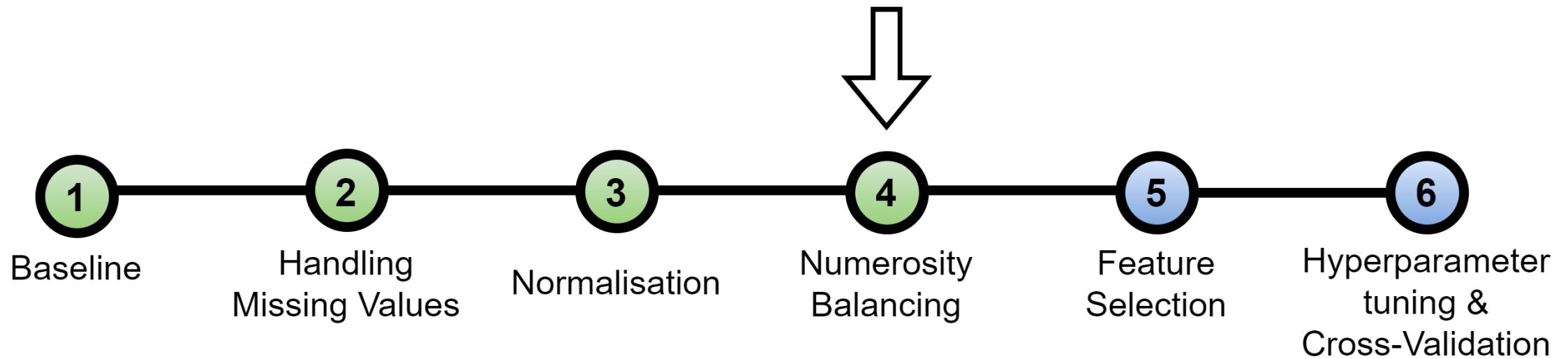


Experimental Results: Normalisation

- Min-max normalisation was applied to deal with large differences in the scales of features
- Results did not differ significantly with the RF classifier and with datasets that previously had essentially no categorical features

Classifier	Dataset	Min-max normalisation	Acc (%)	F1-Score (%)	AUC-ROC (%)
SVM	CICAndMal2017	✗	66.13	78.96	51.51
		✓	70.81	79.71	63.22
	AM	✗	70.23	00.00	50.00
		✓	90.88	82.77	85.89

Experimental Results: Numerosity Balancing

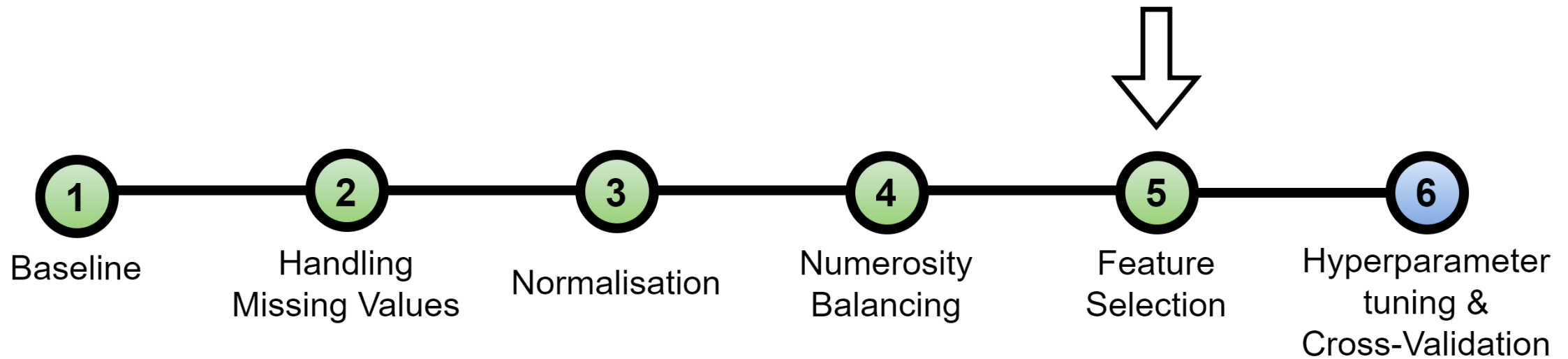


Experimental Results: Numerosity Balancing

- Dealing with numerosity balancing:
 - Use of the AM dataset (the most imbalanced dataset)

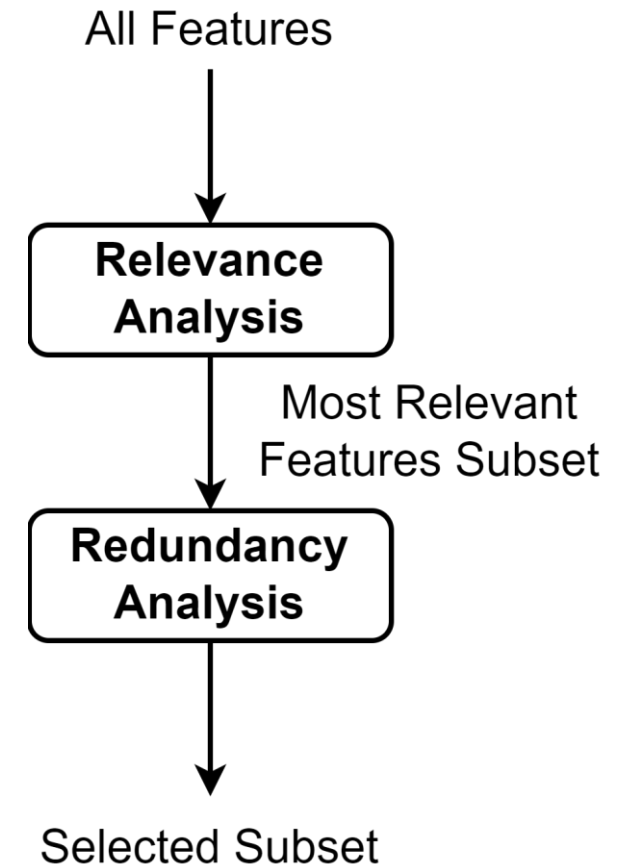
Classifier	Numerosity balancing method	Acc (%)	Rec (%)
RF	None	93.28	83.02
	Random undersampling	91.36	86.44
	Random oversampling	96.28	96.07
	Synthetic Minority Oversampling Technique (SMOTE)	94.06	91.39
SVM	None	70.81	86.00
	Random undersampling	89.18	82.44
	Random oversampling	89.47	82.75
	Synthetic Minority Oversampling Technique (SMOTE)	88.81	81.42

Experimental Results: Feature Selection

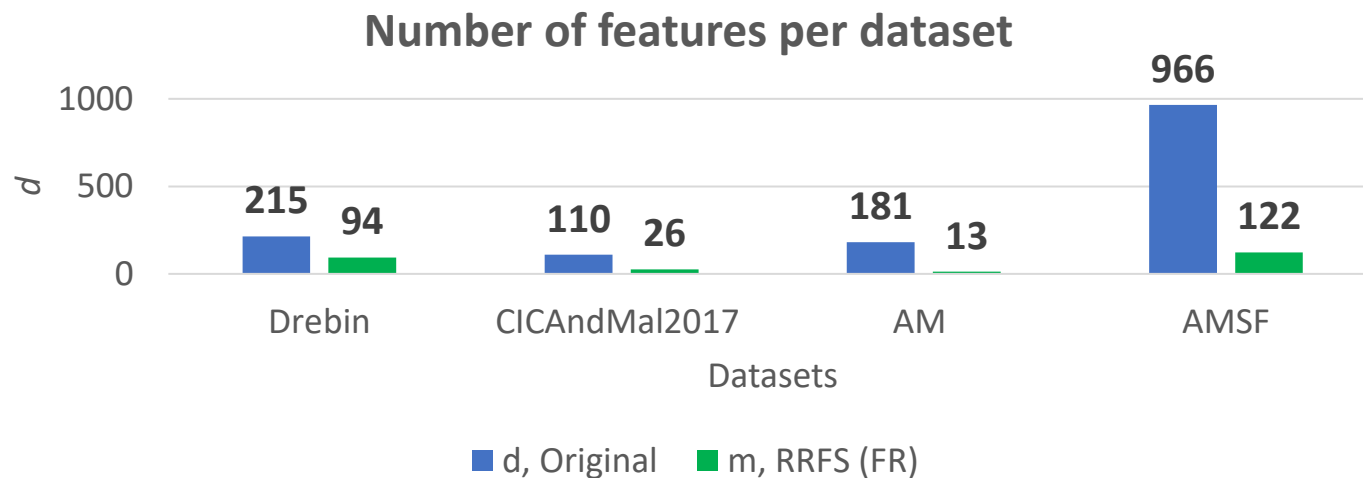
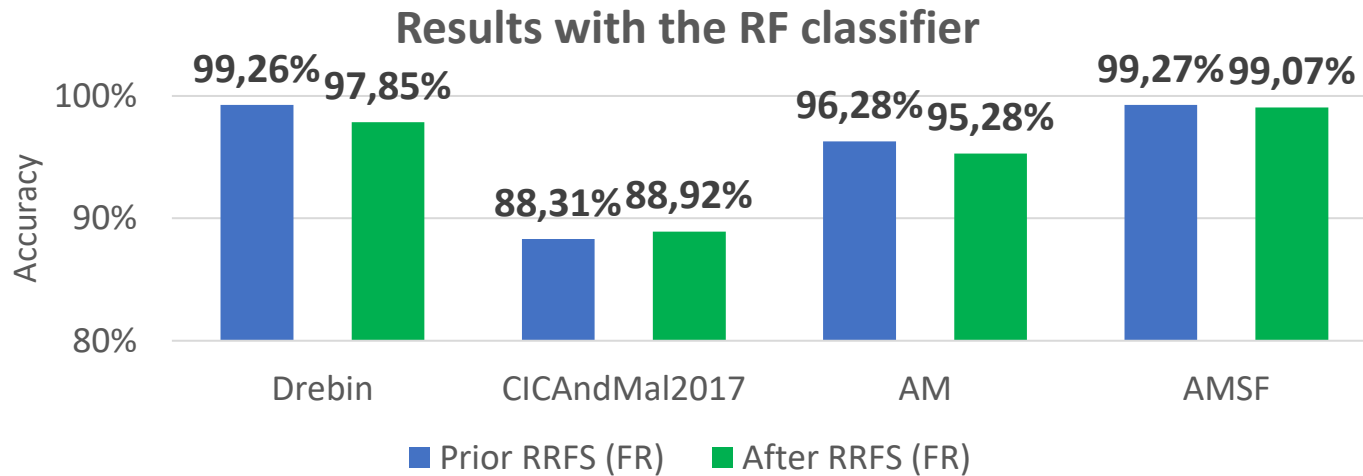


Experimental Results: Feature Selection

- A filter method was applied
- Relevance-Redundancy Feature Selection (RRFS)
 - Fisher ratio (FR) relevance measure (supervised)
 - Mean-median (MM) relevance measure (unsupervised)
 - Overall, FR (supervised) outperformed MM (unsupervised)
 - Thus, the class label data is impactful to the result
 - Absolute cosine (AC) redundancy measure



Experimental Results: Feature Selection (2)



- Substantial dimensionality reduction compensated for a slight metric decrease
- Drebin dataset:
≈56% reduction
- CICAndMal2017 dataset:
≈76% reduction
- AM dataset:
≈93% reduction
- AMSF dataset:
≈87% reduction

Experimental Results: Feature Selection (3)

- 4 most indicative features of malware presence in each dataset

Drebin dataset

1. transact
2. SEND_SMS
3. Ljava.lang.Class.getCanonicalName
4. android.telephony.SmsManager

AM dataset

1. com.android.launcher.permission.UNINSTALL_SHORTCUT
2. android.permission.VIBRATE
3. android.permission.ACCESS_FINE_LOCATION
4. name

CICAndMal2017 dataset

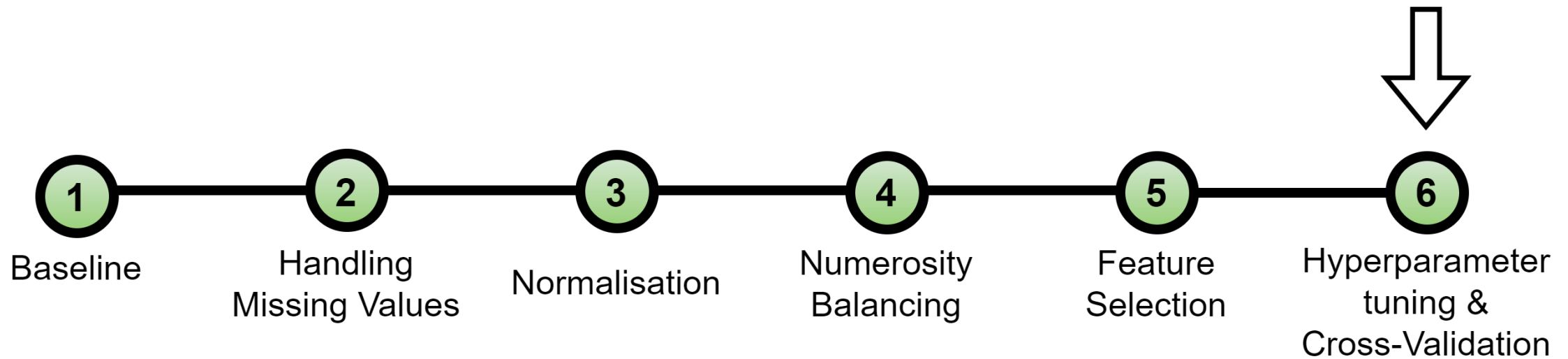
1. Category
2. Price
3. Network communication : view network state (S)
4. Your location : access extra location provider commands (S)

AMSF dataset

1. android.permission.SEND_SMS
2. android.telephony.SmsManager.sendTextMessage
3. float-to-int
4. android.telephony.SmsManager

- Overall, permissions seem to have a prevalent presence among the most relevant features for Android malware detection

Experimental Results: Hyperparameter tuning & CV



Experimental Results: Hyperparameter tuning & CV

- **Hyperparameter tuning**

- Optimisation of the hyperparameters deemed more impactful
- Use of a function that optimises the hyperparameters. It uses 5-fold CV with the training and validation sets
- The metrics results improved slightly (about 2%)

- **Cross-Validation (CV)**

- 10-fold CV and Leave-one-out CV were applied to the training and testing sets, leading to nested CV
- Often challenging due to “training time bottlenecks”
- Mean and standard deviation values for the different metrics
- Standard deviation values were low

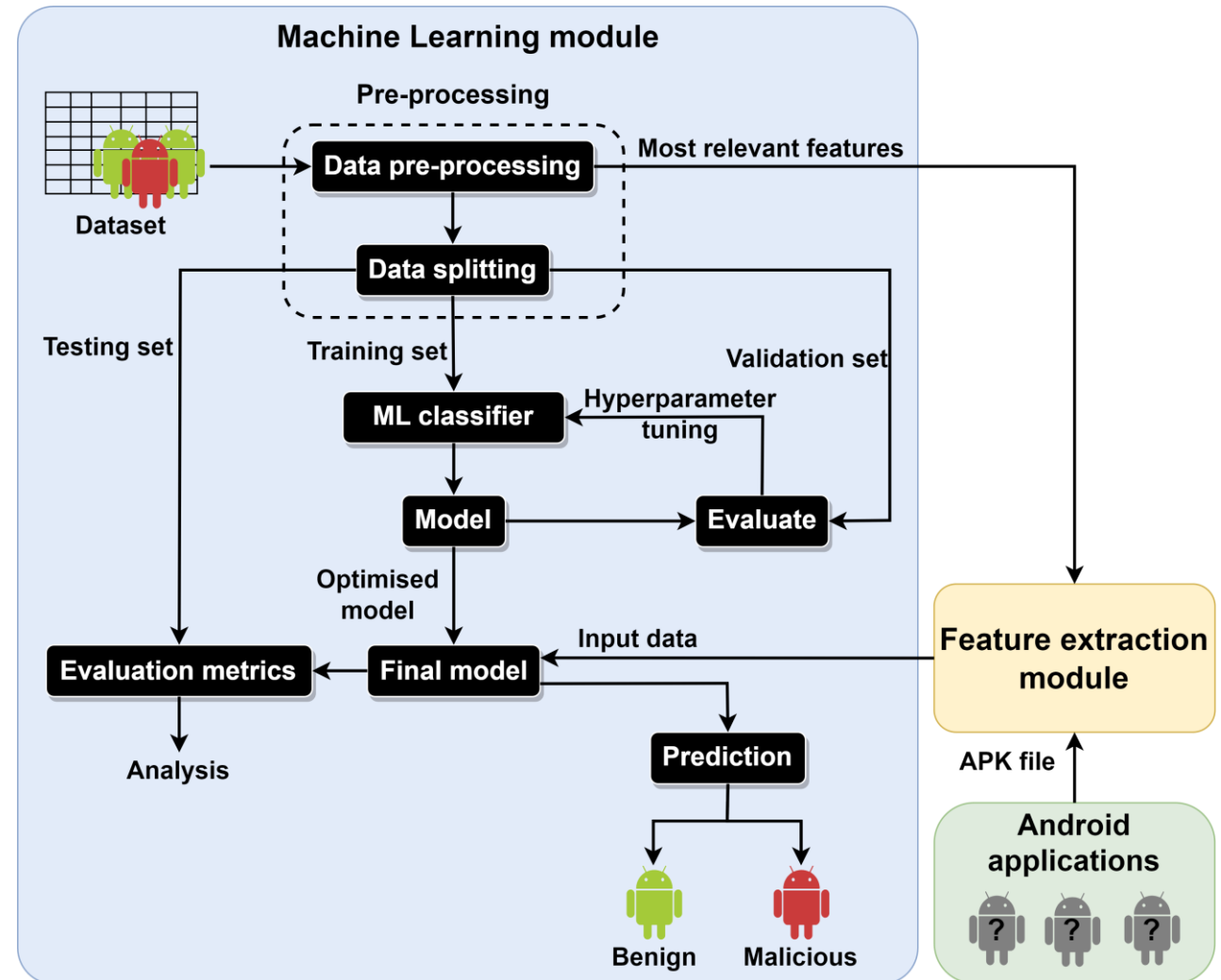
Comparative Analysis of Results

- *'Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices'*, Alkahtani and Aldhyani, 2022

Classifier	Dataset	Accuracy (%)	
		Alkahtani and Aldhyani	Proposed
SVM	Drebin	80.71	97.47
	CICAndMal2017	100.00	73.22

- Data pre-processing greatly impacts the results
- *'Android malware detection applying feature selection techniques and machine learning'*, Keyvanpour *et al.*, 2023
 - The authors applied FS techniques to the Drebin dataset
 - Reported features, SEND_SMS and android.telephony.SmsManager, were also selected on the Drebin and AMSF datasets by RRFS

Complete approach



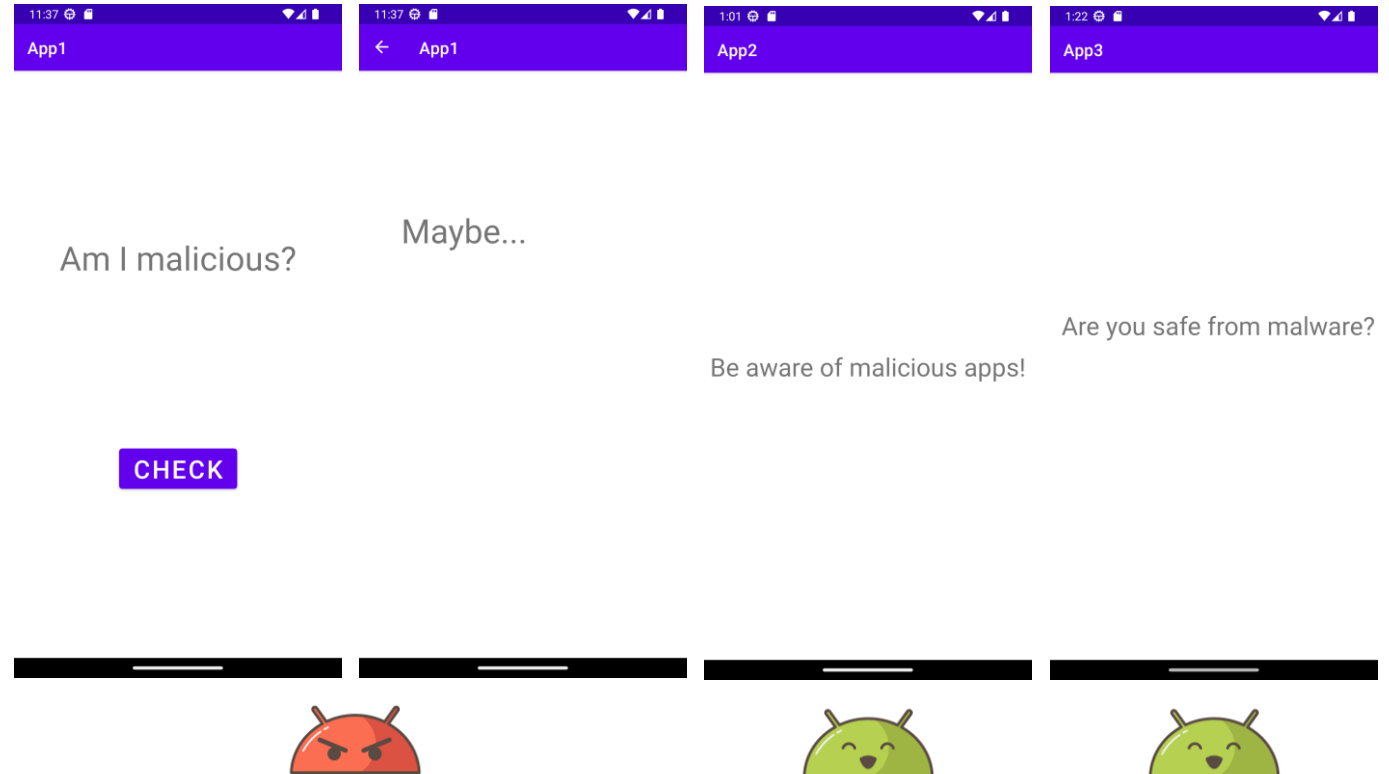
Experimental Results: Real-world Applications

- Simple Android apps built for testing:

- App1 - requests permissions regarding SMS and other features selected as the most relevant in the Drebin and AMSF datasets

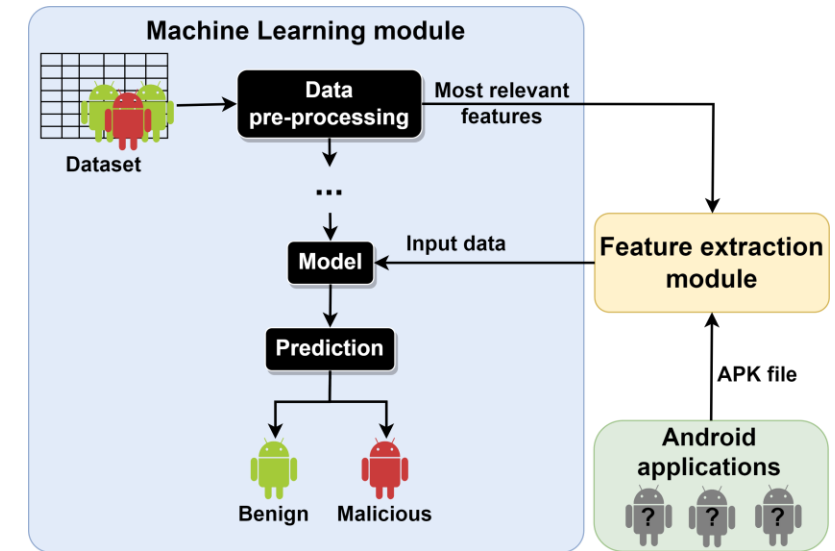
- App2 - doesn't request/use any unnecessary features

- App3 – requests some permissions selected as the most relevant features in the Drebin and AM datasets



Experimental Results: Real-world Applications (2)

- Experiments performed with the RF classifier
- App1 classified as malicious
 - with models trained with the Drebin, CICAndMal2017, and AMSF datasets
- App2 and App3 are classified as benign
 - with models trained with the Drebin, AM, and AMSF datasets
- Experiments performed with other APK found online
- The non-standardization of feature names presents a major challenge
 - Example: 'android.permission.SEND_SMS' \neq 'SEND_SMS' \neq 'androidpermissionSEND_SMS'
(Drebin dataset) (AMSF dataset)



Conclusions

- The RF and SVM classifiers present the best results
- We were able to identify the most relevant features in each dataset for malware detection in Android apps
- Overall, permissions have a prevalent presence among the most relevant features for Android malware detection
- ML and FS approaches effectively mitigate this problem
- No model performs globally best for all datasets
- Use of the ML model in real-world scenarios is not straightforward

Future Work

- Use more up-to-date datasets
- Aim to use datasets more standardised
- Expand the proposed approach to hybrid analysis
- Further explore Deep Learning approaches and others
- Address this problem as multiclass

Contributions



- Catarina Palma, Artur Ferreira, and Mário Figueiredo, "[*On the use of machine learning techniques to detect malware in mobile applications*](#)", Simpósio em Informática (INForum), September 2023, Porto, Portugal



- Catarina Palma, Artur Ferreira, and Mário Figueiredo, "[*A study on the role of feature selection for malware detection on Android applications*](#)", Portuguese Conference on Pattern Recognition (RECPAD), October 2023, Coimbra, Portugal



- Catarina Palma, Artur Ferreira, and Mário Figueiredo, "[*Explainable Machine Learning for Malware Detection on Android Applications*](#)", Information journal, MDPI, January 2024



- Public [GitHub repository](#) for the code developed in the context of the thesis



Contributions (2)

ON THE USE OF MACHINE LEARNING TECHNIQUES TO DETECT MALWARE IN MOBILE APPLICATIONS

Catarina Palma¹
A45241@alunos.isel.pt

Artur Ferreira^{1,3}
artur.ferreira@isel.pt

Mário Figueiredo^{1,3}
mario.figueiredo@tecnico.ulisboa.pt

¹ ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
² IST, Instituto Superior Técnico, Universidade de Lisboa
³ IT, Instituto de Telecomunicações, Lisboa

The Problem – Malware in Mobile Apps

- 70% of mobile phones use Android
- In Q3 2022, Google Play Store hosted around 3.5 million apps
- Android applications are a prized target for malware developers
- Existing security measures to mitigate malware are, to some extent, successful
- However, malware keeps growing in both sophistication and diffusion
- In 2020, 5.7 million Android malware packages were detected, tripling 2019's 2.1 million

Public Domain Datasets

	Drebin	CICAndMal2017
Instances	15036	29999
Features	215	183
Release year	2014	2018
Categorical features	✓	✓
Numerical features	✓	✓
Missing values	✓	✓
Class label ratio	1/3	1/3
Class label majority	benign	malicious

Goals

- Explore machine learning (ML) and feature selection (FS) approaches to detect malware in Android apps
- Check the importance and impact of:
 - data pre-processing
 - feature selection
 - different classification techniques

Proposed Approach

- Supervised ML approach
- Binary classification problem

Techniques and Evaluation Metrics

Data pre-processing

- Categorical features → numerical features, through label encoding
- Different methods to impute missing values
- Min-Max normalisation

Feature Selection

- Relevance-redundancy FS (RRFS)
- Fisher ratio relevance measure (supervised)
- Absolute cosine redundancy measure

Data splitting

- Random split
- 70/30 ratio for train/test

ML classifiers

- Support Vector Machine (SVM)
- K-Nearest Neighbours (KNN)
- Naive Bayes (NB)

Evaluation Metrics

- Confusion Matrix**
 - True positive (TP) → malicious app as malicious
 - True negative (TN) → benign app as benign
 - False positive (FP) → benign app as malicious
 - False negative (FN) → malicious app as benign
- Accuracy (Acc)** = $\frac{TP+TN}{TP+FP+FN+TN}$
- Recall (Rec)** = $\frac{TP}{TP+FN}$ (true positive rate or sensitivity)

Experimental Results and Evaluation

Baseline

Classifier	Dataset	Acc (%)	Pre (%)	FN	FP	Rec (%)	
RF	Drebin	98.80	2834	13	50	1634	97.03
RF	CICAndMal2017	80.49	2000	930	781	5001	86.49
SVM	Drebin	97.94	2885	22	71	1633	95.76
SVM	CICAndMal2017	65.82	6	2984	14	5768	99.76
KNN	Drebin	97.58	2782	45	64	1620	96.20
KNN	CICAndMal2017	64.00	940	2050	1108	4672	80.84
NB	Drebin	93.08	2611	216	96	1588	94.30
NB	CICAndMal2017	65.30	461	2529	497	5285	91.40

Handling Missing Values

Classifier	Dataset	Acc (%)	Pre (%)	FN	FP	Rec (%)	
RF	Drebin	80.53	2074	938	790	4982	86.13
RF	CICAndMal2017	80.88	2089	883	838	5190	86.30
RF	Drebin	81.06	2088	884	811	5207	86.38
SVM	Drebin	65.74	219	2771	234	5548	95.95
SVM	CICAndMal2017	67.28	419	2052	892	5638	93.50
SVM	Drebin	67.07	273	2599	365	5683	93.94

Feature Selection

Classifier	Dataset	Acc (%)	Baseline	Acc (%)	RRFS
RF	Drebin	98.60	98.80	96.92	
RF	CICAndMal2017	80.49	81.42		
SVM	Drebin	97.94	96.36		
SVM	CICAndMal2017	65.82	70.42		

Number of Features for each Dataset

Dataset	# d (Original)	# d (after RRFS)
Drebin	215	94
CICAndMal2017	183	64

Conclusions

- ML and FS approaches effectively mitigate this problem
- RF and SVM classifiers present the best results
- The baseline and dimensionality-reduced datasets exhibit similar metrics
- Results arguably compensated by dimensionality reduction
- A reduction of 56% in the Drebin dataset and 65% in the CICAndMal2017 dataset
- No ideal solution was found

Future Work

- Further investigation with different FS techniques
- Additional experiments with different datasets
- More evaluation metrics should be considered

RECPAD 2023

A STUDY ON THE ROLE OF FEATURE SELECTION FOR MALWARE DETECTION ON ANDROID APPLICATIONS

Catarina Palma¹
A45241@alunos.isel.pt

Artur Ferreira^{1,3}
artur.ferreira@isel.pt

Mário Figueiredo^{1,3}
mario.figueiredo@tecnico.ulisboa.pt

¹ ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
² IST, Instituto Superior Técnico, Universidade de Lisboa
³ IT, Instituto de Telecomunicações, Lisboa