It Works on My Machine: A Systematization of Knowledge on Reproducibility and Replicability in ML-Based Ransomware Detection

Bruno Pereira, Beatriz Cepa, Cláudia Brito, João Marco Silva, Vera Rimmer, João Paulo, Tânia Esteves

KU LEUVEN

INESC TEC

Introduction | Cryptographic Ransomware

Encryption techniques used to block access to files

Demands a ransom in exchange for restoring the system

Responsible for high profile attacks resulting in huge financial losses

The rising threat continues to push researchers to act!

Introduction | Cryptographic Ransomware

Detection is one approach followed by researchers to **stop attacks**

ML has proven **useful** to **detect complex patterns** and analyse large datasets

ML has been used in **many papers/works** to help **find** ransomware **behavioral patterns**

Introduction | Cryptographic Ransomware

Detection is one approach followed by researchers to **stop attacks**

ML has proven **useful** to **detect complex patterns** and analyse large datasets

ML has been used in **many papers/works** to help **find** ransomware **behavioral patterns**

Can we validate and (re)use existing solutions?

Study | Methodology

Objective: Assess reproducibility and replicability in ML-based ransomware detection

Analyzed **32 papers** evaluating **38 different criteria** in each of them

- Proposed solution must be evaluated with **ransomware samples**
- Must propose an **AI-based classification or detection** for cryptographic ransomware
- Must incorporate **behavioral analysis** and use storage I/O features

Analyze ... of them

- P ... **re samples**
- M ... **n** for
- M ... e I/O

TABLE 2. CLASSIFICATION OF ML-BASED RANSOMWARE DETECTION RESEARCH BASED ON ITS REPRODUCIBILITY AND REPLICABILITY.

	Dataset			Tested Samples										Behavioral Analysis							Feature Extraction				ML Training				Availability		
				Sample Size			Ransomware					Benign										Input	Techniques	Output					Dataset		Code
Papers	Sample	Behavioral	Feature	# Ransomware	# Malware	# Benign	# Families	Families Name & Distribution	Collection Date	Source	Samples Hash	Apps class	Apps name	Similar RW behavior	Source	Environment	Operating System	Network	File System	User Behavior	Analysis Tools	Execution Time	Groups	Operations	Feature Engineering	Feature Selection	Data Type	Data Format	Train/Test set	Model Types	Libraries/frameworks (Hyper-)parameters	Explainability	Sample	Behavioral	Feature	Pseudo-code	Artifacts
[48]	○	○	○	582	—	942	11	✓	✓	✓	+/-	✓	✗	✓	✓	S	W XP	✓	✓	✗	AP	✓	✓	+/-	K	K	B	✓	✓	ST	✓	✓	✗	✗	✓	✗	✗	✗
[14]	○	○	○	688	—	2245	11	✓	✓	✓	+/-	✗	✗	✓	✗	SR	W 7, 8.1, 10	✓	✓	✓	C	✓	✓	✓	K	—	N	✓	✓	ST	✗	+/-	✗	?	✓(*)	?	PC	✓(*)
[25]	○	○	○	360	532	460	21	✓	✗	✓	✗	✗	✗	✗	✗	S	W XP W 7	✗	✗	✗	AP	✗	✓	✗	✗	K	✗	✗	✓	ST	✓	+/-	✗	✗	✗	✗	✗	✗
[28]	○	○	○	1624	—	220	3	✓	✓	✓	✗	✗	✗	✗	✓	S	W 10	✗	✗	✓	C	✓	✓	✓	KC	K	N	✓	✓	MT	✓	✗	✗	✗	✗	✗	PC	+/-
[12]	○	○	○	83	—	85	4	+/-	✗	✓	✗	✓	✗	✗	✓	S	W 7	✗	✗	✗	MT	✓	✓	✗	KC	K	N	✓	✓	MT	✓	✗	✗	✗	✗	✗	✗	✗
[35]	○	○	○	157	—	?	✗	✗	✗	✓	✗	✗	✗	✗	✗	S	W ?	✗	✗	✗	AP	✓	✓	✗	K	—	N	✓	✗	ST	✓	✓	✗	✗	✗	✗	✗	+/-
[23]	◐	○	○	1928	—	2500	8	✓	✓	✓	✗	✗	✗	✗	✓	S	A 4.1	✗	✗	✓	C	✗	✓	✗	K	—	N	✗	✓	ST	✓	✗	✗	✗	✗	✗	✗	+/-
[45]	○	○	○	38152	—	1000	?	✗	✓	✓	✗	✗	✗	✗	✓	S	W XP	✗	✗	✗	AP	✓	✓	✗	K	K	N	✓	✓	ST	✓	✗	✗	✗	✗	✗	PC	✗
[36]	○	○	○	14	—	?	14	+/-	✗	✓	✗	✗	✓	✓	✗	S	W ?	✗	✗	✗	MT	✗	✓	✓	✗	✗	N	✓	✗	MT	✗	✗	✗	✗	✗	✗	✗	✗
[10]	○	○	○	10	—	15	8	✓	✗	✓	✗	✓	✓	✓	✗	S	I	✗	✗	✗	✗	✗	+/-	+/-	C	—	N	✗	✗	ST	✗	✗	✗	✗	✗	✗	✗	✗
[51]	○	○	○	276	—	312	?	✗	✗	✓	✗	✓	✗	✗	✓	S	W 7	✗	✓	✗	AP	✗	✓	✗	K	—	N	✓	✓	ST	✗	✗	✗	✗	✗	✗	F	✗
[22]	◐	○	○	672	—	2386	5	+/-	✓	✓	✗	✗	✗	✗	✓	S	A 4.0	✗	✗	✓	MT	?	✓	✗	✗	✗	N	✓	✓	MT	✗	✗	✗	✗	✗	✗	F	✗
[38]	○	○	○	✗	—	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	I	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	ST	✗	✗	✗	✗	✗	✗	✗	✗
[9]	○	○	○	942	830	292	✗	✗	✗	✗	✗	✗	✗	✗	✓	S	W 7	✗	✗	✗	MT	✓	✓	✗	KC	—	N	✓	✓	MT	✓	✗	✗	✗	✗	✗	F	✗
[8]	○	○	○	*	—	*	✗	✗	✗	✓	✗	✗	✗	✓	✗	S	W 7	✗	✗	✗	AP	✗	✓	✗	K	K	N	✓	✗	MT	✗	+/-	✗	✗	✗	✗	F	✗
[11]	○	○	○	7	—	✗	7	✓	✓	✗	✗	✗	✗	✗	✗	S	W ?	✗	✗	✓	AP	✗	✗	✗	K	K	N	✓	✗	ST	✓	✗	✗	✗	✗	✗	✗	✗
[19]	○	○	○	200	—	200	?	✗	✗	✓	✗	✗	✗	✗	✓	S	W 10	✓	✗	✗	AP	✗	✗	✗	✗	✗	N	✓	✗	MT	✓	✓	✗	✗	✗	✗	✗	+/-
[24]	○	○	○	67	—	?	?	✗	✗	✓	✗	✗	✗	✓	✗	✗	W ?	✗	✗	✗	✗	✗	+/-	✗	✗	✗	✗	✓	✗	ST	✗	✗	✗	✗	✗	✗	✗	✗
[33]	—	●	○	582	—	942	11	+/-	✗	—	—	✓	✗	✗	—	—	W ?	—	—	—	—	—	✓	✗	✗	C	✗	✓	✓	ST	✗	✗	✗	—	—	✗	PC	✗
[41]	○	○	○	8152	—	1000	15	✓	✓	✓	✗	✓	✗	✓	✓	S	W ?	✗	✗	✗	AP	✗	✓	✗	KC	KC	N	✓	✓	MT	✓	✗	✗	✗	✗	✗	PC	✗
[1]	—	●	○	582	—	942	11	✓	✗	—	—	✗	✗	✗	—	—	W ?	—	—	—	—	—	✓	✗	✗	K	B	✓	✓	MT	✓	✓	✗	—	—	✗	✗	✗
[30]	○	○	○	668	—	103	21	✓	✗	✓	✗	✗	✗	✓	✓	S	W 7	✓	✓	✗	AP	✓	✓	✗	K	K	NB	✓	✓	MT	✓	✗	✗	✗	✓*	✗	PC	✗
[29]	○	○	○	*	—	*	✗	✗	✗	✗	✗	✗	✗	✗	✗	S	W ?	✗	✗	✗	AP	✗	✓	+/-	✗	✗	N	✓	✓	ST	✗	✗	✗	✗	✗	✗	✗	✗
[56]	○	○	○	35015	500	500	14	✓	✓	✓	✗	✗	✗	✗	✓	✗	W ?	✗	✗	✗	✗	✗	✓	✗	C	—	B	✓	✓	MT	✓	✗	✗	✗	✗	✗	PC	✗
[49]	○	○	○	483	—	754	14	✓	✓	✓	✗	✗	✗	✗	✓	S	W ?	✗	✗	✗	AP	✓	✓	+/-	K	K	B	✓	✓	ST	✓	✓	✗	✗	✗	✗	PC	✗
[3]	○	○	○	673	—	742	14	✓	✓	✓	✗	✓	✓	✓	✓	S	W XP	✓	✓	✓	AP	✓	✓	✗	K	K	B	✓	✓	ST	✗	✗	✗	✗	✗	✗	PC	✗
[4]	○	○	○	1354	—	1358	14	✓	✗	✓	✗	✓	✗	✓	✓	S	W XP	✓	✓	✓	AP	✓	✓	+/-	K	C	✗	✓	✓	MT	✗	✗	✗	✗	✗	✗	PC	✗
[55]	○	○	○	899	—	241	9	✓	✓	✓	✗	✗	✗	✗	✗	S	W ?	✗	✗	✗	AP	✗	✓	✗	K	—	N	✓	✓	ST	✓	✓	✗	✗	✗	✗	✗	✗
[52]	○	○	○	1761	—	1768	?	✗	✓	✓	✗	✓	✗	✗	✗	S	W ?	✗	✗	✗	✗	✗	+/-	+/-	C	—	✗	✓	✗	ST	✗	✗	✗	✗	✗	✗	PC	✗
[53]	○	○	○	13	—	16	10	✗	✗	✗	✗	✗	✓	✓	✗	?	WS 16, 19, 22, C	✗	✗	✗	C	✗	+/-	+/-	✗	✗	✗	✗	✓	MT	✗	✗	✗	✗	✗	✓	✗	✗	✗
[50]	—	●	○	383	—	?	5	✓	✗	—	—	✗	✗	✗	—	—	—	—	—	—	—	—	✓	✓	K	—	✗	✗	✓	ST	✓	✗	✗	—	—	✓	✗	✓
[27]	○	◐	○	476	—	2245	10	✓	✓	✗	✗	✓	✗	✓	—	S	W 11	✗	✓	✗	✗	✗	+/-	+/-	✗	✗	N	✓	✓	ST	✗	✗	✓	✗	✗	✗	✗	✗

Global
— Not applicable
? - Unclear
* - Inconsistent
✓- Provided
✗- Not provided
+/- - Partially provided

Dataset
○ New dataset
◐ Partially new
● Reuse dataset

Environment & OS
S - Sandbox
R - Real
W - Windows
WS - Windows Server
A - Android
C - CentOS

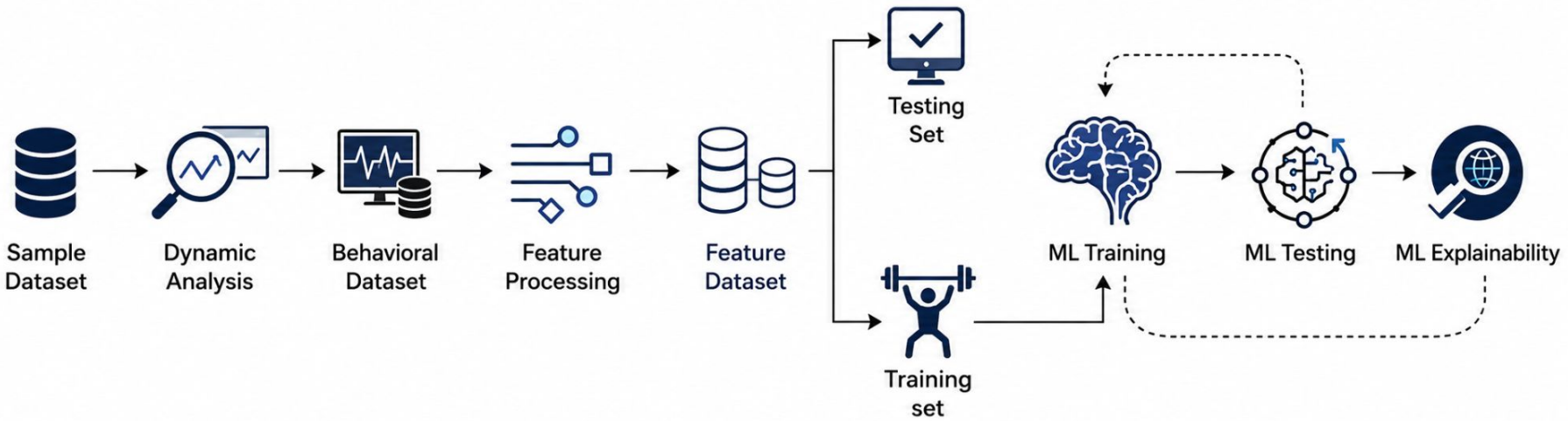
Analysis Tool
AP - Analysis Platform
MT - Monitoring Tool
C - Custom

Feature
K - Known
C - Custom
B - Binary
N - Numeric

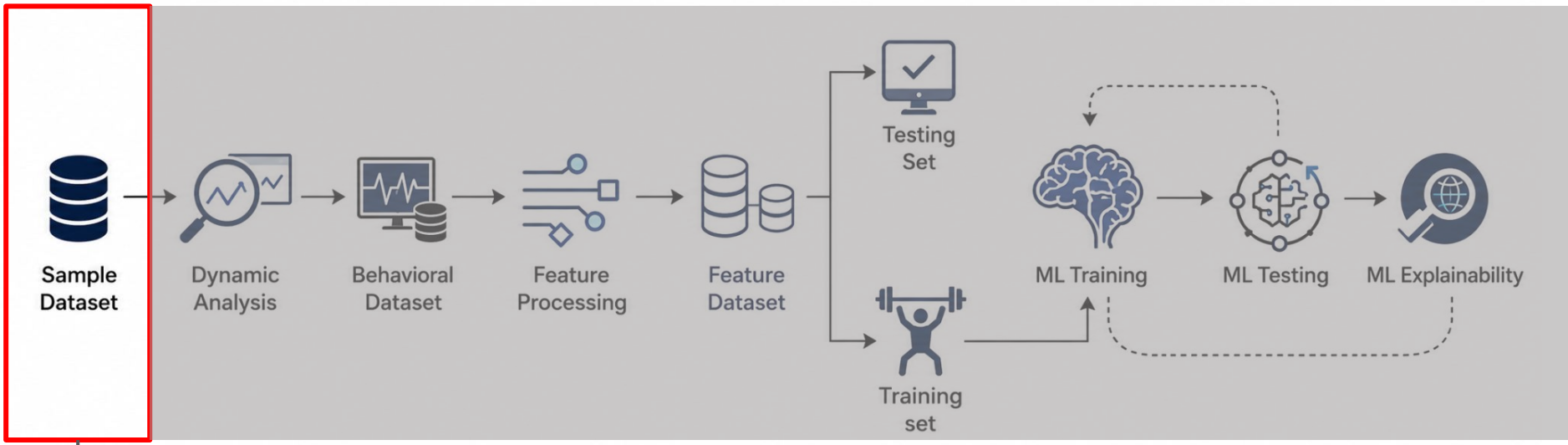
Model Types
ST - Single
MT - Multiple

Code
PC - Pseudo-code
F - Formulas
✓(*) - Unavailable

Study | Development Pipeline



Study | Development Pipeline



- Collection of representative sample sets to train and test the detection solution

Study | Sample Dataset

- What do we need to know?

  Sample Distribution: number of benign and malware samples

  Dataset composition: ransomware families and benign applications used
  - ➔ Ransomware Families (malware hash)
  - ➔ Benign applications and their versions

  Sources: references to malware and benign software repositories

Study | Sample Dataset

- What do we need to know?

Sample Distribution: number of benign and malware samples

Table 2: Statistics of the collected low-level I/O data from 383 ransomware samples.

Ransomware Family	No. Samples	Data	#IRPs Millions
CryptoWall	157 (41.0%)	8.0	286.7
Crowti	125 (32.6%)	5.7	173.1
CryptoDefense	77 (20.1%)	4.5	171.6
Critroni	14 (3.7%)	0.6	3.0
TeslaCrypt	10 (2.6%)	0.9	29.2
Total	383	19.7	663.6

Detailed description of a ransomware dataset.

[1] Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., & Maggi, F. (2016, December). Shieldfs: a self-healing, ransomware-aware filesystem. In Proceedings of the 32nd annual conference on computer security applications

Study | Sample Dataset

- What do we need to know?

Sample Distribution: number of benign and malware samples

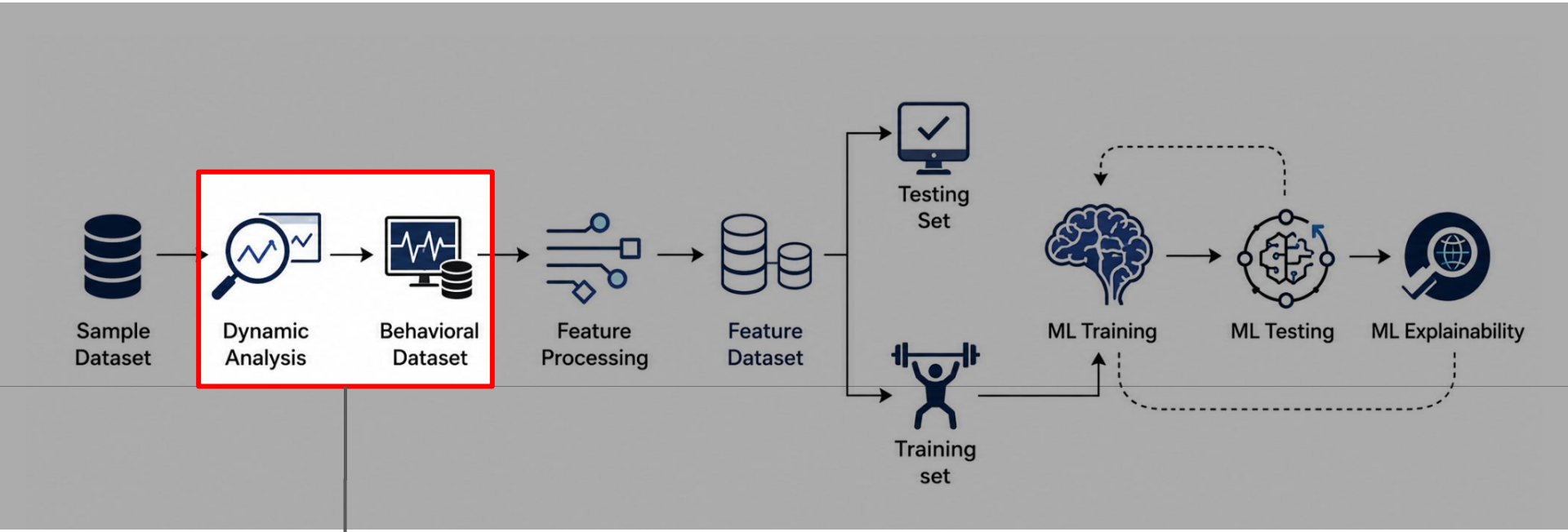
Dataset composition: ransomware families and benign applications used

> *"Our Dataset contains about 1.7 billion IRP produced by **2245 different applications** [1]."*

We do not know which applications were used.
Impossible to reproduce

[1] Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., & Maggi, F. (2016, December). Shieldfs: a self-healing, ransomware-aware filesystem. In Proceedings of the 32nd annual conference on computer security applications

Study | Development Pipeline



Execution of samples and collection of traces

Study | Behavioral Dataset

Reproducibility checklist

Analysis Environment

Hardware Specification, Virtualization, Sandbox Technology

File System State

Structure, Files, Directories

Network Configuration

Connectivity, blocked / allowed

Operating System

Name and version

User Activity Simulation

Mouse clicks, browsing

Monitoring / Analysis

Tools Used and Configuration

Study | Behavioral Dataset

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

Operating System

File System State

User Activity Simulation

Network Configuration

Monitoring / Analysis

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Study | Behavioral Dataset

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

Operating System

File System State

Normal User Files (My documents, My pictures and Videos, etc)

User Activity Simulation

Network Configuration

Monitoring / Analysis

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Study | Behavioral Dataset

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

File System State

Normal User Files (My documents, My pictures and Videos, etc)

Network Configuration

Virtualbox Machine with controlled Access to internet **host-only adapter**

Operating System

User Activity Simulation

Monitoring / Analysis

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

File System State

Normal User Files (My documents, My pictures and Videos, etc)

Network Configuration

Virtualbox Machine with controlled Access to internet **host-only adapter**

Operating System

Host: **Ubuntu 16.05 LTS**
Guest: **Windows XP server pack 3 32bits**

User Activity Simulation

Monitoring / Analysis

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

File System State

Normal User Files (My documents, My pictures and Videos, etc)

Network Configuration

Virtualbox Machine with controlled Access to internet **host-only adapter**

Operating System

Host: **Ubuntu 16.05 LTS**
Guest: **Windows XP server pack 3 32bits**

User Activity Simulation

Custom python script performs user activity such as browsing, clicking and deleting documents

Monitoring / Analysis

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Ahmed et al. [3]

Analysis Environment

Cuckoo Sandbox

File System State

Normal User Files (My documents, My pictures and Videos, etc)

Network Configuration

Virtualbox Machine with controlled Access to internet **host-only adapter**

Operating System

Host: **Ubuntu 16.05 LTS**
Guest: **Windows XP server pack 3 32bits**

User Activity Simulation

Custom python script performs user activity such as browsing, clicking and deleting documents

Monitoring / Analysis

Cuckoo Sandbox

[3] Yahye Abukar Ahmed et al. “A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection”. In: Journal of Network and Computer Applications 167 (2020)

Study | Behavioral Dataset

Hasan et al. [4]

- Analysis Environment
- Operating System
- File System State
- User Activity Simulation
- Network Configuration
- Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

Operating System

File System State

User Activity Simulation

Network Configuration

Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

File System State

Network Configuration

Operating System

User Activity Simulation

Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

File System State

Network Configuration

Operating System

User Activity Simulation

Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

File System State

Network Configuration

Operating System

Host: **Ubuntu 12.04**
Guest: **Windows XP and 7**

User Activity Simulation

Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

Operating System

Host: **Ubuntu 12.04**
Guest: **Windows XP and 7**

File System State

User Activity Simulation

Network Configuration

Monitoring / Analysis

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Behavioral Dataset

Hasan et al. [4]

Analysis Environment

Cuckoo Sandbox

Operating System

Host: **Ubuntu 12.04**
Guest: **Windows XP and 7**

File System State

User Activity Simulation

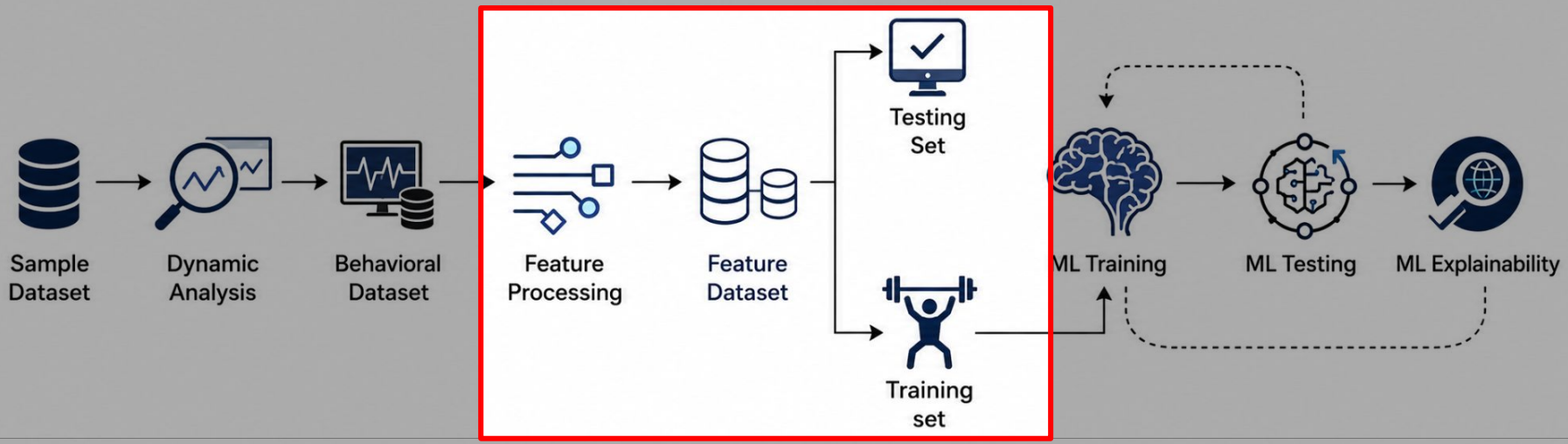
Network Configuration

Monitoring / Analysis

Cuckoo Sandbox

[4] Hasan, M. M., & Rahman, M. M. (2017, December). RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th international conference of computer and information technology (ICCIT) (pp. 1-7). IEEE.

Study | Development Pipeline



Study | Feature Dataset

Reproducibility checklist

Input Data

Type of data collected, Specific data retained

Preprocessing

Tools, scripts and thresholds used

Feature Engineering

Detailed process description, techniques used, parameters

Feature Selection

Methods used, number of features selected

Output Dataset

Structure of the dataset, data type, file format, dimensions

Study | Feature Dataset

Ahmed et al. [5]

Execution traces collected in a sandbox environment

Only **API function call names** are extracted from the traces

N-grams used to generate features

Enhanced Maximum-Relevance and Minimum-Redundancy algorithm applied

Top 90 features selected

[5] Ahmed, Y. A., Koçer, B., Huda, S., Al-Rimy, B. A. S., & Hassan, M. M. (2020). A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. Journal of Network and Computer Applications, 167, 102753.

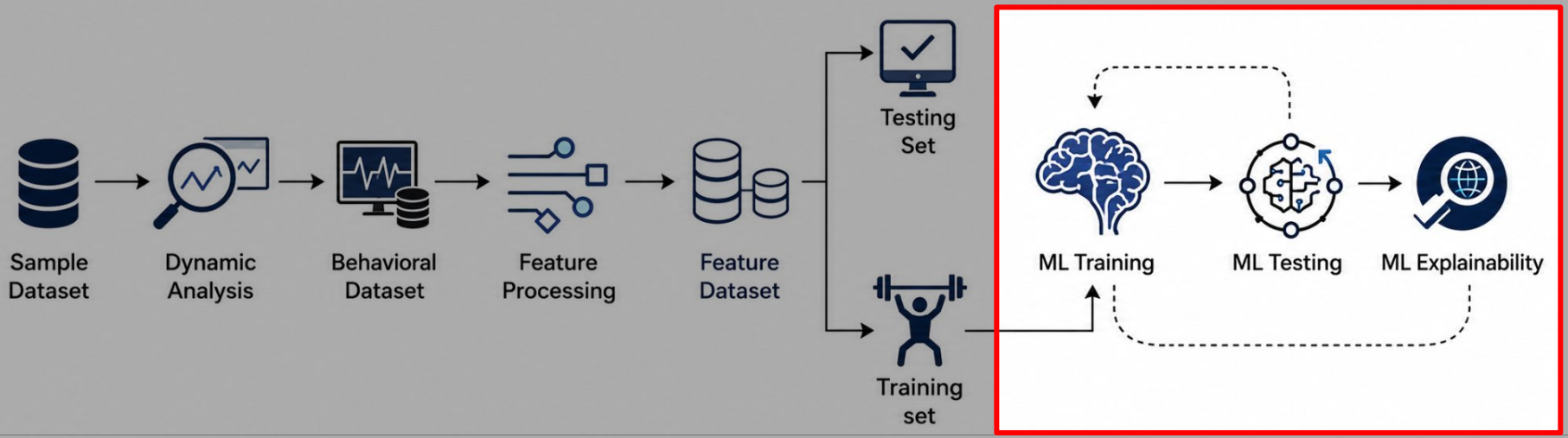
Study | Feature Dataset

Ahmed et al. [5]	Ferrante et al. [6]
Execution traces collected in a sandbox environment	Execution logs collected in a sandbox environment
Only **API function call names** are extracted from the traces	
N-grams used to generate features	
Enhanced Maximum-Relevance and Minimum-Redundancy algorithm applied	
Top 90 features selected	In total 87 features were extracted

[5] Ahmed, Y. A., Koçer, B., Huda, S., Al-Rimy, B. A. S., & Hassan, M. M. (2020). A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. Journal of Network and Computer Applications, 167, 102753.

[6] Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F., & Milosevic, J. (2017, October). Extinguishing ransomware-a hybrid approach to android ransomware detection. In International symposium on foundations and practice of security (pp. 242-258). Cham: Springer International Publishing.

Study | Development Pipeline



Reproducibility checklist

Training

Model types, Hyperparameters, random seeds, optimization, hardware, libraries/versions

Testing

Metrics (precision, recall, F1)

Explainability

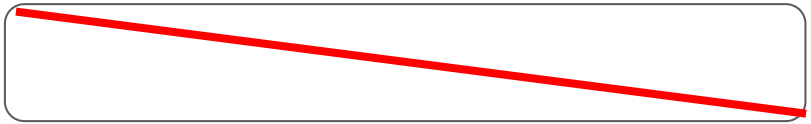
Use explainbility techniques to validate the model

Study | Machine Learning Development

Zhou et al. [7]

SVM model used, with hyperparameters reported in the paper

Different metrics reported, F1, recall, precision and support



[7] Zhou, J., Hirose, M., Kakizaki, Y., & Inomata, A. (2020, February). Evaluation to Classify Ransomware Variants based on Correlations between APIs. In ICISSP (pp. 465-472)

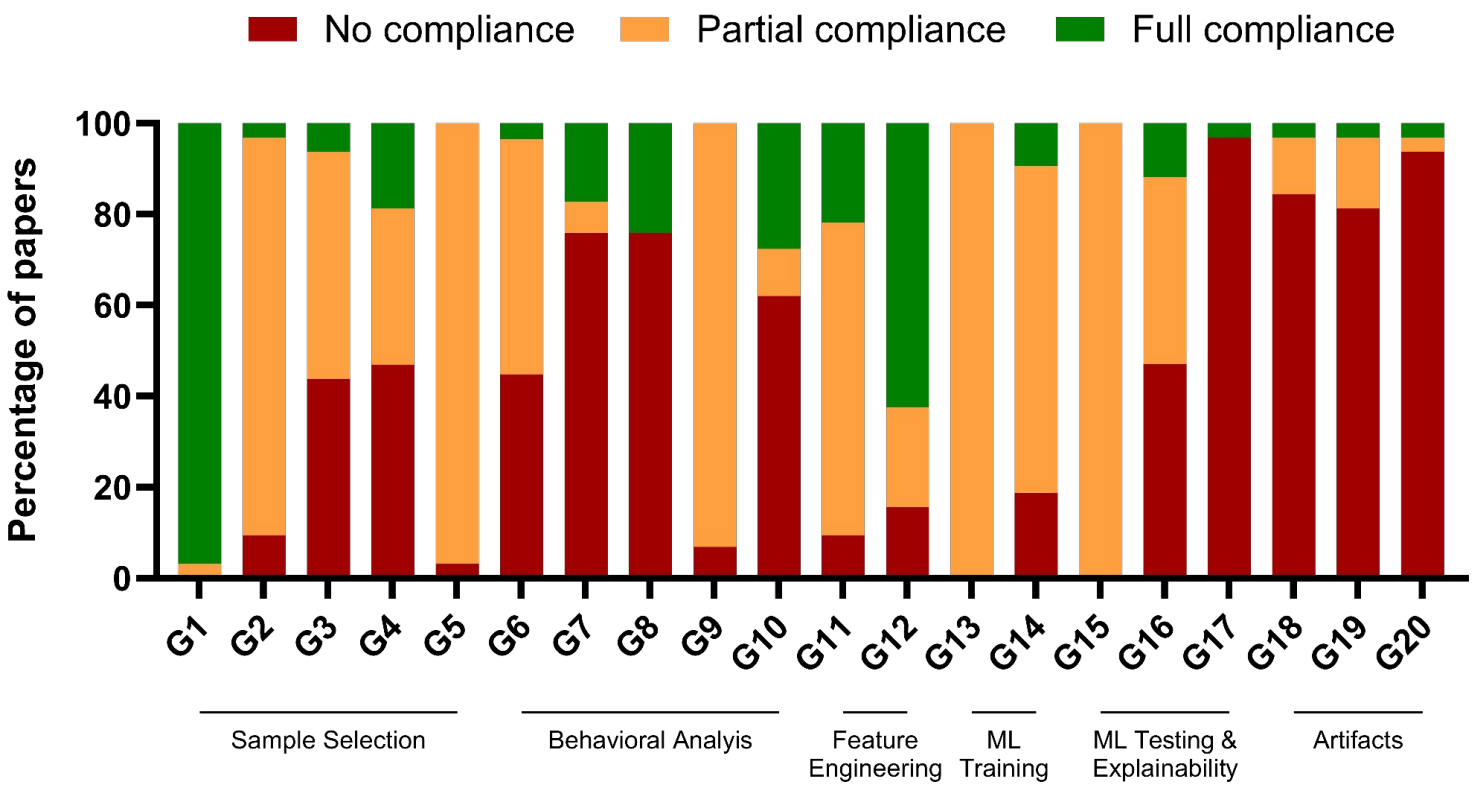
Study | Machine Learning Development

Zhou et al. [7]	Wang et al. [8]
SVM model used, with hyperparameters reported in the paper	Gradient Boosting Decision Tree via XGBoost, **no hyperparameters**
Different metrics reported, F1, recall, precision and support	True / False Positives / Negatives

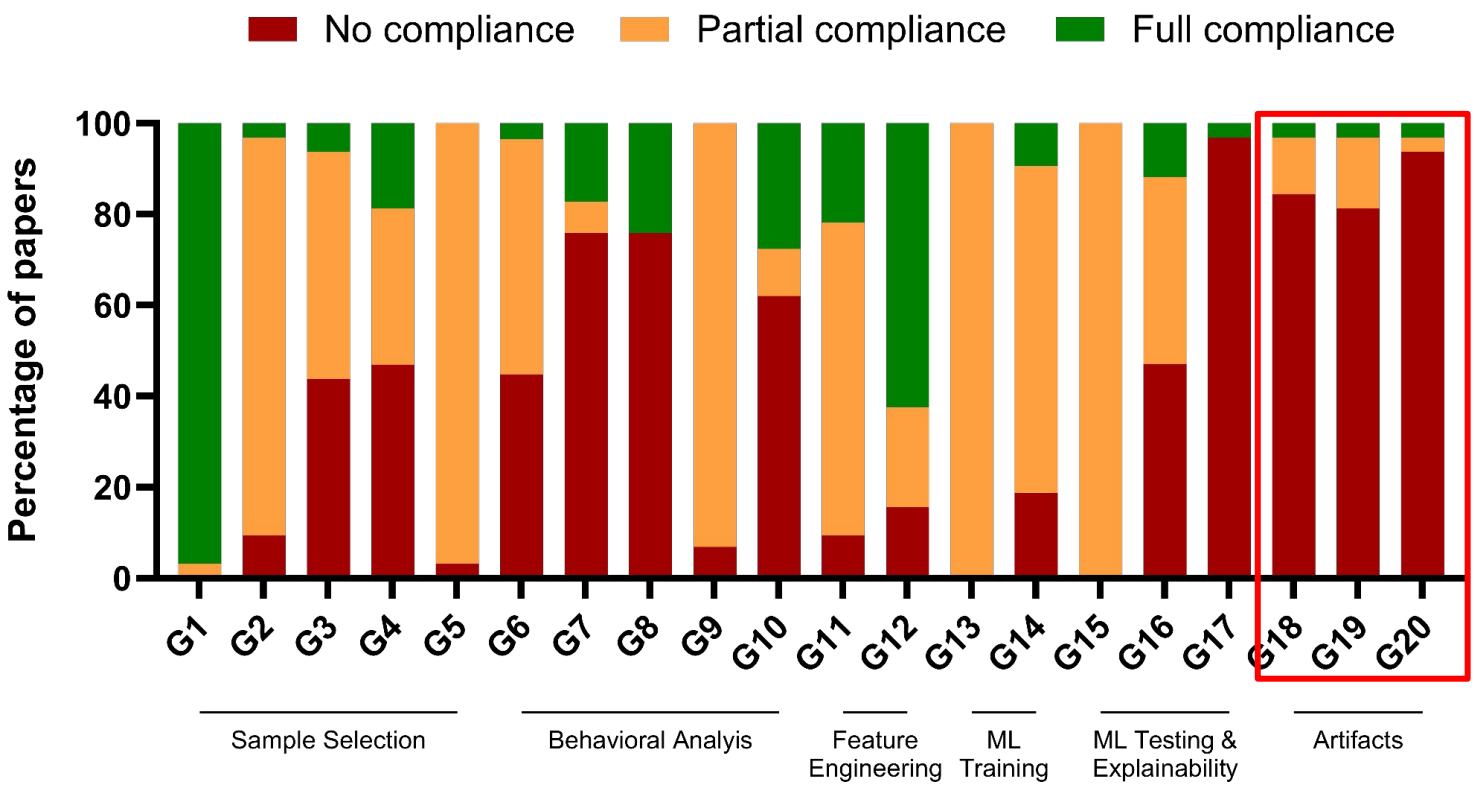
[7] Zhou, J., Hirose, M., Kakizaki, Y., & Inomata, A. (2020, February). Evaluation to Classify Ransomware Variants based on Correlations between APIs. In ICISSP (pp. 465-472)

[8] Wang, S., Dong, F., Yang, H., Xu, J., & Wang, H. (2024, December). Cancal: Towards real-time and lightweight ransomware detection and response in industrial environments. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (pp. 2326-2340).

Discussion


No compliance
Partial compliance
Full compliance
Percentage of papers
100
80
60
40
20
0
G1
G2
G3
G4
G5
G6
G7
G8
G9
G10
G11
G12
G13
G14
G15
G16
G17
G18
G19
G20
Sample Selection
Behavioral Analyis
Feature Engineering
ML Training
ML Testing & Explainability
Artifacts

Discussion


No compliance
Partial compliance
Full compliance
Percentage of papers
100
80
60
40
20
0
G1
G2
G3
G4
G5
G6
G7
G8
G9
G10
G11
G12
G13
G14
G15
G16
G17
G18
G19
G20
Sample Selection
Behavioral Analyis
Feature Engineering
ML Training
ML Testing & Explainability
Artifacts

Discussion | Future Work

Proposed guidelines are **useful** for the community to develop new **reproducible research**

Following all of the guidelines can prove **challenging and time consuming**

Explore how the different **stages of the pipeline** can be **automated** and **optimized**

Automation would **increase** the **reproducibility** and validation of future research

It Works on My Machine: A Systematization of Knowledge on Reproducibility and Replicability in ML-Based Ransomware Detection

Bruno Pereira, Beatriz Cepa, Cláudia Brito, João Marco Silva, Vera Rimmer, João Paulo, Tânia Esteves

https://dsr-haslab.github.io/

KU LEUVEN

INESC TEC