CLOUD BASED MALICIOUS PDF DETECTION USING MACHINE LEARNING

Viorel GURDIS

TABLE OF CONTENTS

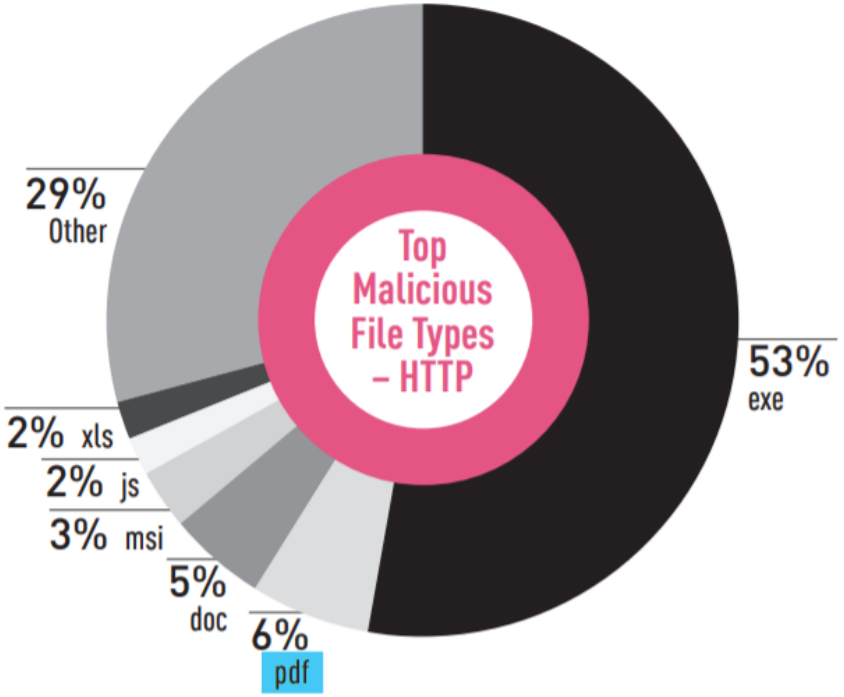
ARE PDF FILES TRUSTWORTHY?

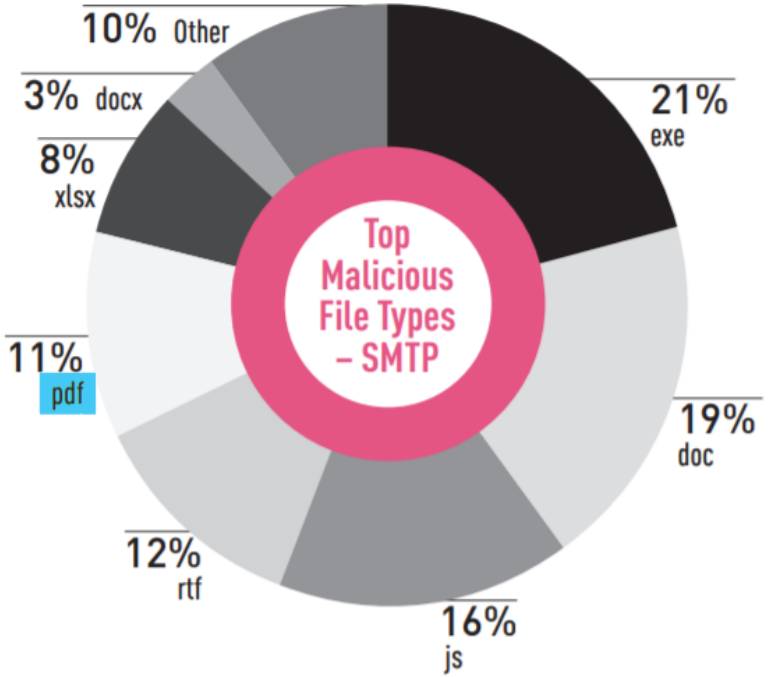
PDF Readers Vulnerabilities

Used for Phishing Attacks

Embedded unverified JavaScript code

PDF IN TOP OF WELL-KNOWN THREATS

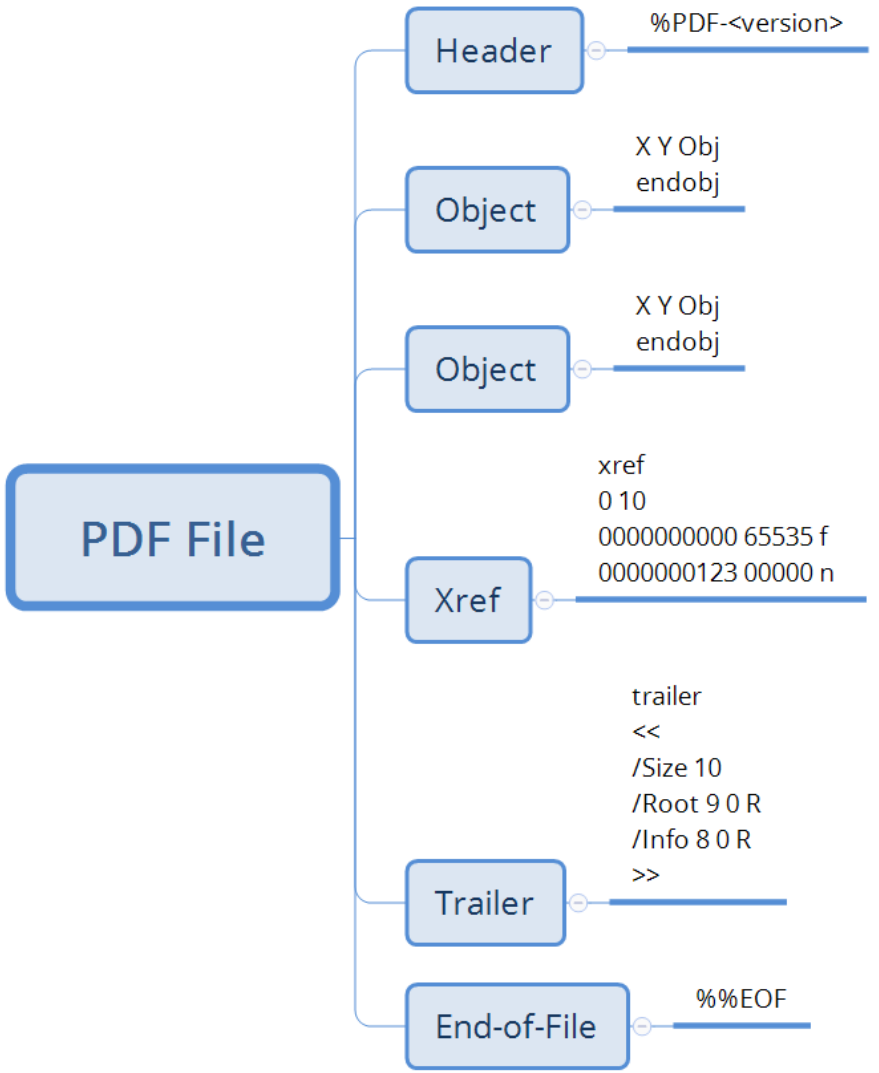




Check Point Research. Cyber Attack Trends: 2019 Mid-Year Report

PDF FORMAT STRUCTURE

- Developed by Adobe in 1990
- Environment Independent
- Aimed to present text documents, including images, URLs, interactive widgets (e.g. Diagrams, Buttons etc.), 3D models and many others.



MISUSE OF FEATURES

> /Javascript - *Sets JavaScript code to be executed*

> /OpenAction, /Names, /AcroForm, /Action, /AA - Defines a script or an action to be automatically run

> /Launch - Runs a program or opens a document

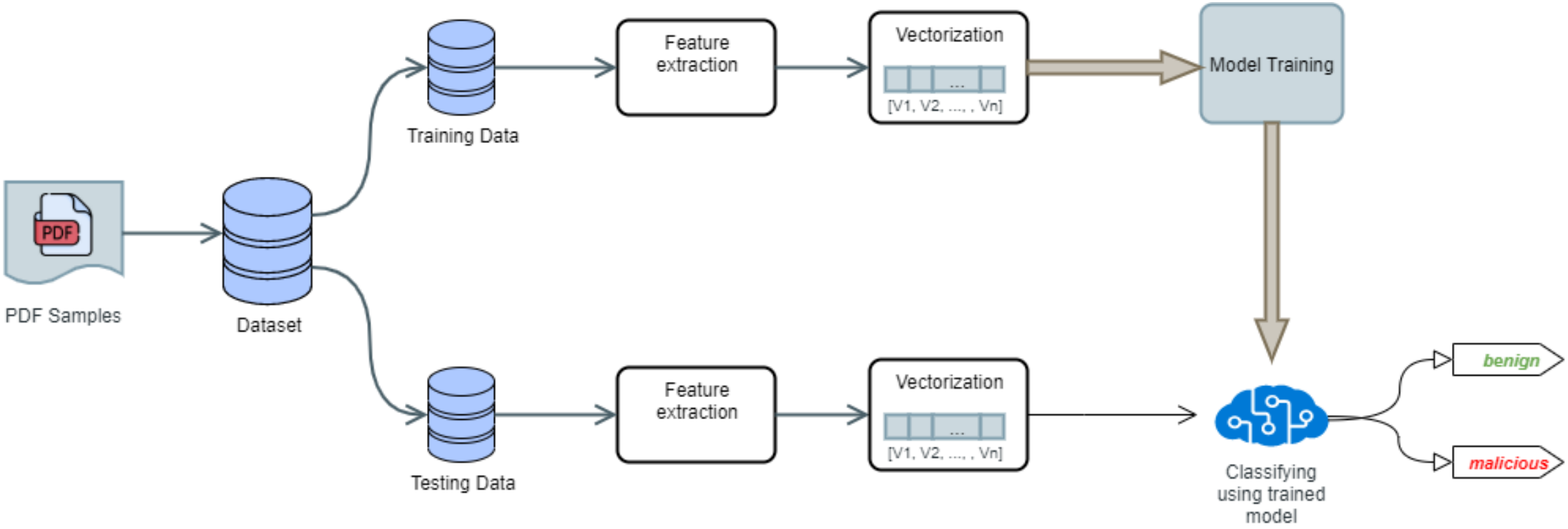
> /URI - Accesses a resource by its URL

> /SubmitForm, /GoToR - Can send data to indicated URL

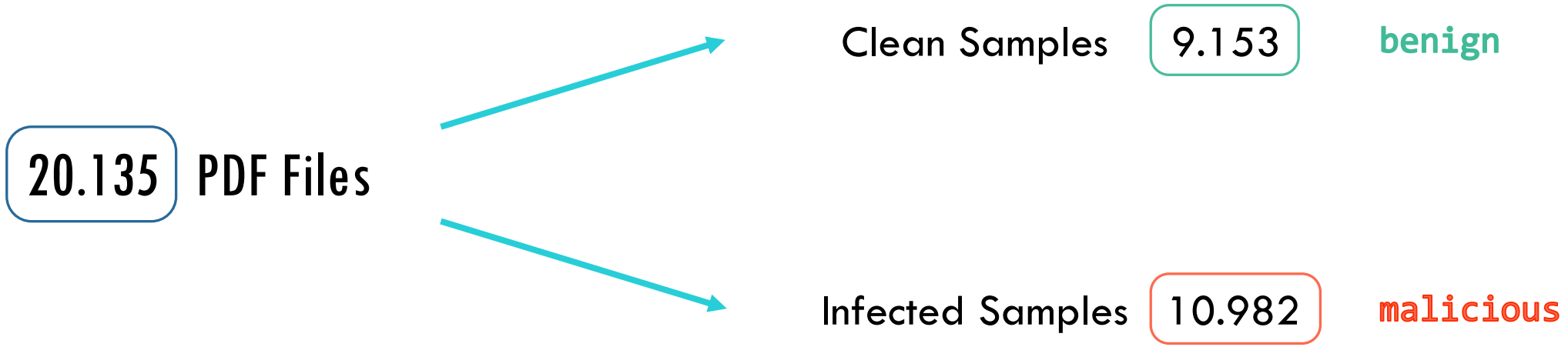
> /ObjStm - Hides objects inside Streams

Obfuscation & Encryption

PROPOSED APPROACH



DATASET



Sources: contagio malware dump, VirusTotal, HYBRID ANALYSIS, TETRA 4D, pdfscripting.com

FEATURE SELECTION

1. **PDFiD** (Didier Stevens' PDF Tools) ⟶ parses a PDF file and counts the occurrences of vulnerable PDF entries, as well as occurrences of name obfuscations.

2. Transform a PDF file into its vectorized form ⟶ Min-Max Normalization on 22 extracted features.

MODEL TRAINING

Dataset Split:

Training Data – 70 %

Testing Data – 30 %

Classification Algorithm – Random Forest (n_estimators = 100)

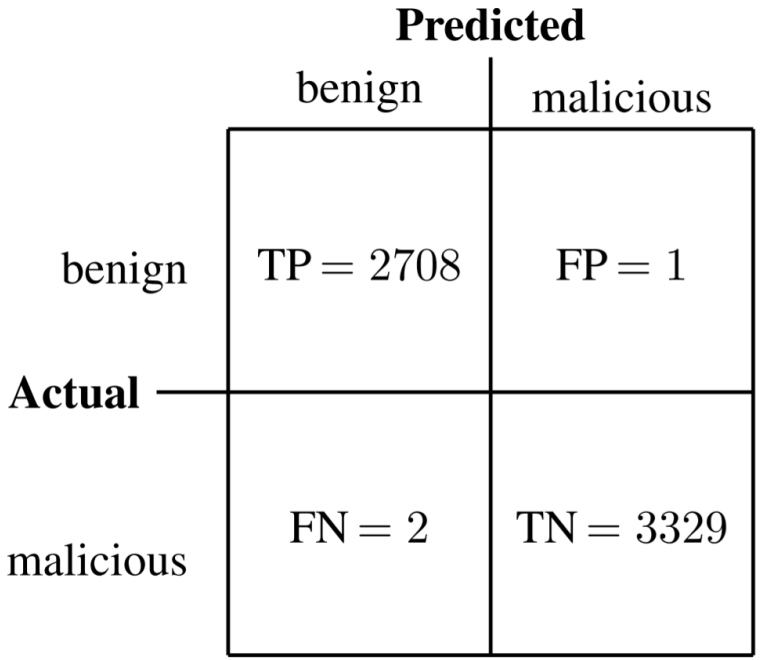
* Training process in an isolated environment, using a Virtual Machine

RESULTS

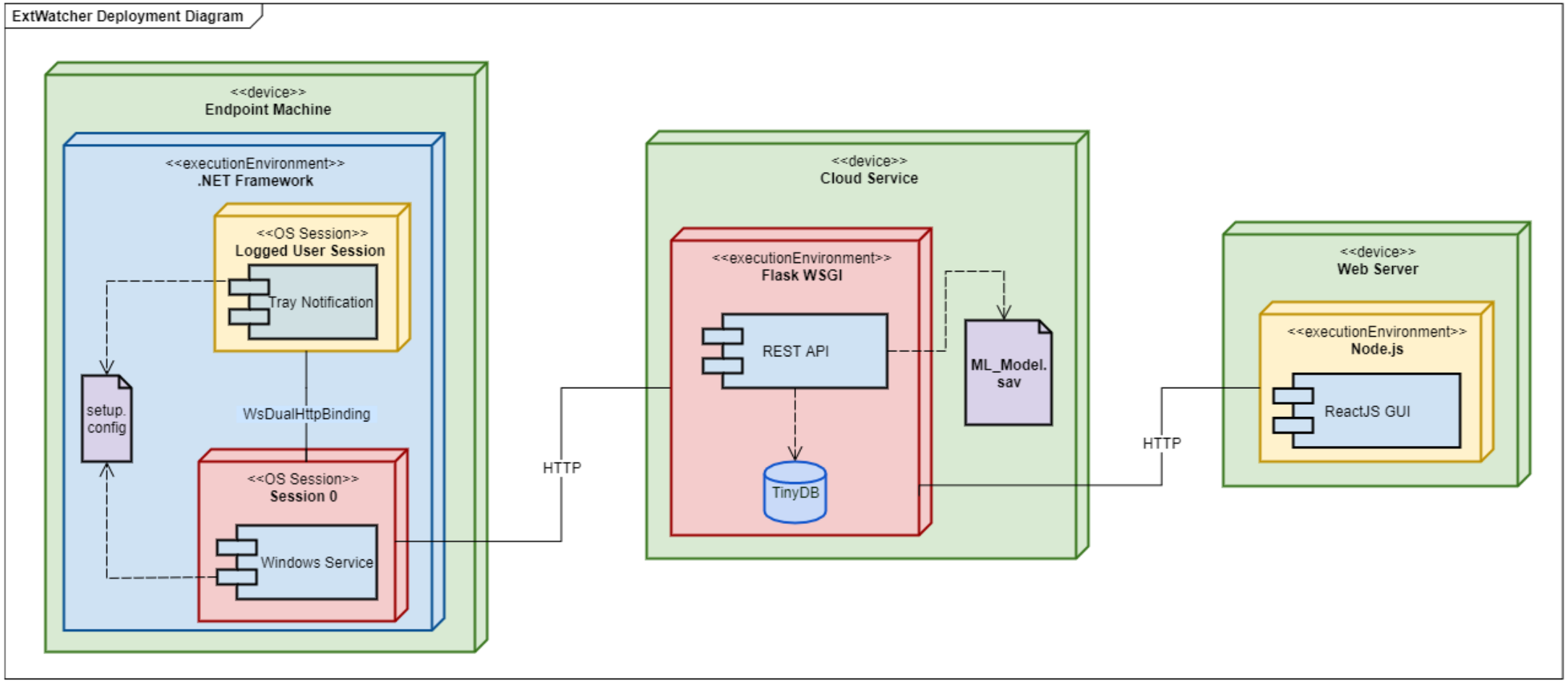
Accuracy = 99,95 %

F1-Score = 99,94 %

Operation	Time
Feature Extraction	29 min
Model Training	0.405 sec
Classification	0.07 sec



EXTWATCHER


ExtWatcher Deployment Diagram
<<device>>
Endpoint Machine
<<executionEnvironment>>
.NET Framework
<<OS Session>>
Logged User Session
Tray Notification
setup.
config
WsDualHttpBinding
<<OS Session>>
Session 0
Windows Service
HTTP
<<device>>
Cloud Service
<<executionEnvironment>>
Flask WSGI
REST API
ML_Model.
sav
TinyDB
HTTP
<<device>>
Web Server
<<executionEnvironment>>
Node.js
ReactJS GUI

ADVANTAGES

- Centralized updates (model retrained only on the Cloud Server)
- Assured privacy of the personal data
- Remote file scanning = isolated secure environment; minimal requirements for user's computer hardware
- Automatic detection for Windows users
- Access to the analyzer via browser (crossplatform support)
- Extendable API

CONCLUSIONS

- The obtained results demonstrate that Machine Learning could handle the task of detecting malicious PDF files.
- As part of the future work we plan to train and integrate Machine Learning models for detecting malware in different file formats, such as: DOC, XLS, PPT, EXE, DLL.

Q & A

Thank you