NFIQ 2

Introducing tools to retrain the classifier

Version 5

About HID (formerly Crossmatch is part of HID)



HID Global powers the trusted identities of the world's people, places and things

DISCLAIMER: Be careful what you wish for!

- When adapting the NFIQ 2, potential interoperability problems needs to be addressed!

NFIQ 2 ≠ NFIQ 2

- Revised NFIQ 2 contains enhancements to mitigate interoperability problems (classifier model as part of the version number)
- The interoperability risk still exists!

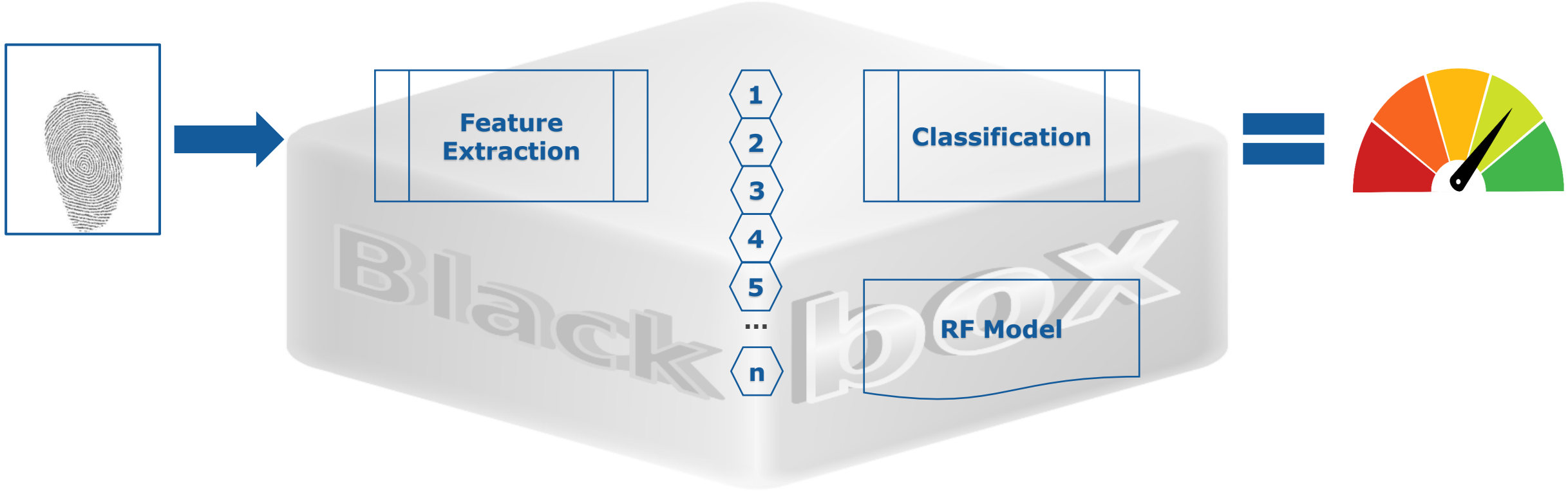
How does a quality assessment work?

- Before we discuss improvements, we need to understand how it works
- Need to open the “black box”
- Understand the architecture of NFIQ 2

Our goal: providing tools for advanced users

This slide uses artificial prints!

Inside NFIQ 2

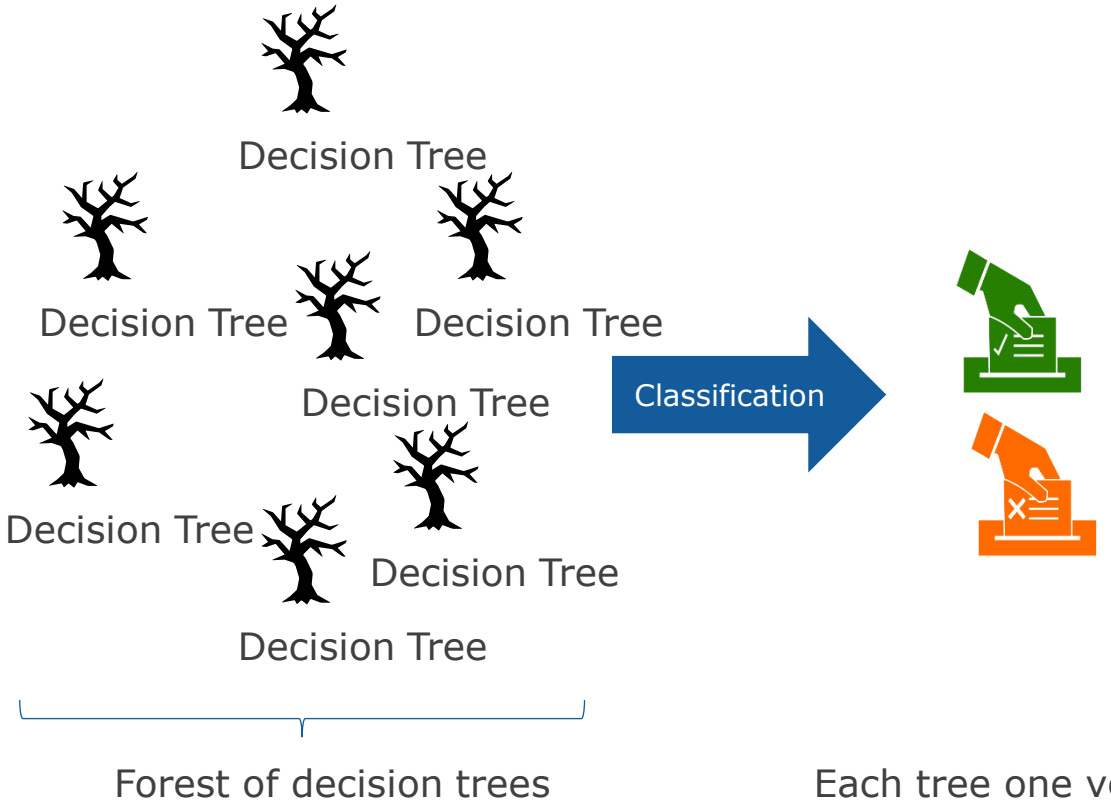


- NFIQ 2 uses 69 features which were manually selected and validated
- The classification is done with a random forest model
- The tools will allow to retrain the classifier
- The selected features will be kept unchanged

This slide uses artificial prints!

Inside the Random Forest

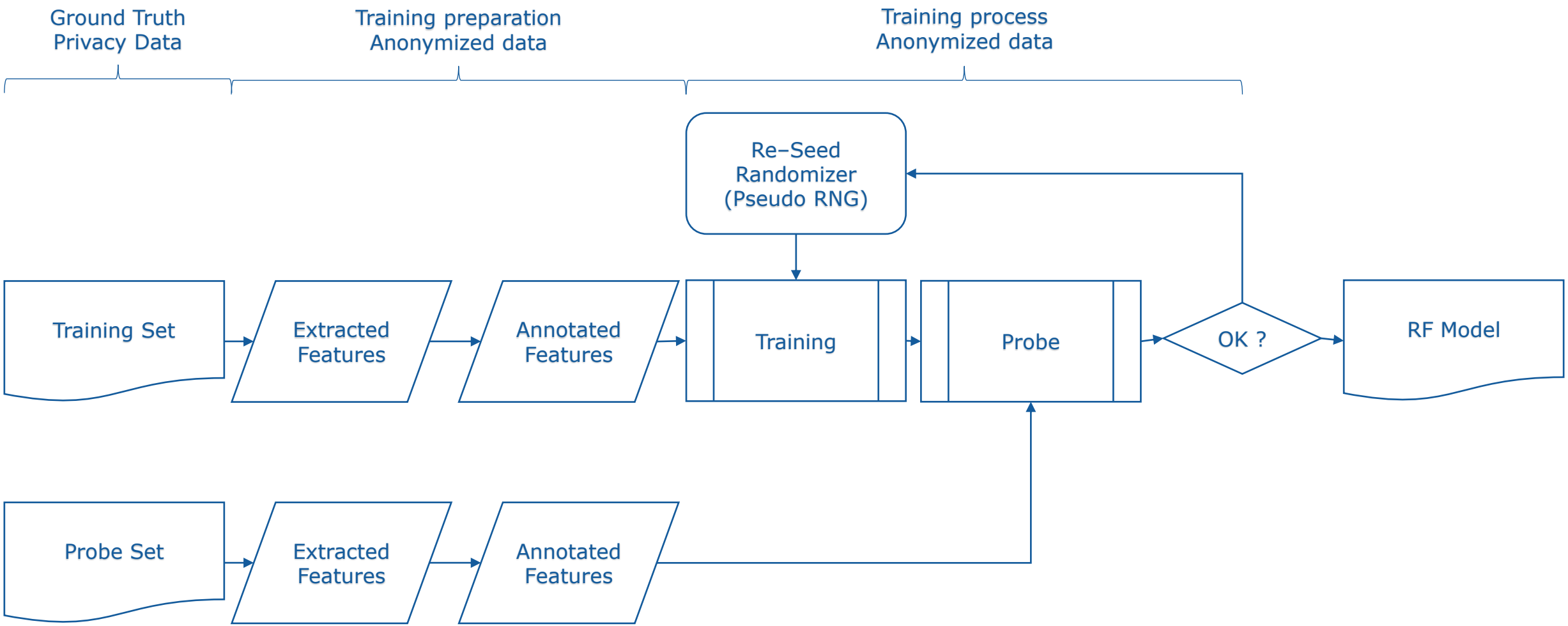
- Random forest is a collection of decision trees (forest)
- Each decision tree uses a random subset of the feature vector (feature bagging)
- Every decision tree has one vote for the final classification



NFIQ 2 random forest parameters

- Binary classification
- 100 decision trees (score values from 0 … 100)
- Active features per decision tree 10 (out of 69)

The training process



Training tools inventory

- Feature extraction
  - Command line executable “nfiq2-calc-features.exe”
  - Using the same function of NFIQ 2, but without using classification
  - Result: semicolon separated string with the sorted feature values (CSV)

- Input preparation and feature annotation
  - Out of scope, must be provided by the operator
  - Sample implementation based on a shell script file
    - Appending the ground truth classification (0,1) to the CSV feature string from the feature extraction
    - Collect all CSV strings into a text file (line by line)

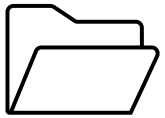
Adapt as needed

- Training and Probing
  - Command line executable “nfiq2-train-classifier.exe”
  - Performs iterative operation
    - Train
    - Probe
    - Repeat if convolution matrix contains errors

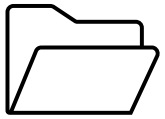
Source code repository (GIT)

- Using the original NFIQ2 from NIST as Git Submodule to de-couple dependencies
- Link to the NFIQ2-Extended Repository
  - Accessible for Biometric Experts of ISO/IEC SC37

Git Repository NFIQ2-Extended



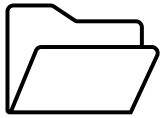
build-scripts
Convenient shell scripts for building



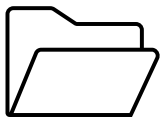
cli-example
Source code for the command line executables

mingw-std-threads
Git Submodule
Origin: https://github.com/meganz/mingw-std-threads.git

NFIQ2Core
Git Submodule
Origin: https://github.com/usnistgov/NFIQ2.git



NFIQ2Training
Source code for the training process

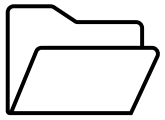


sample-data
Artificial sample data for testing

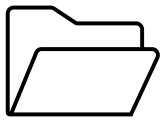
Build artefacts

- The build artefacts will also contain the libraries from the NFIQ2 core build

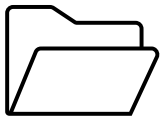
Git Repository NFIQ2-Extended



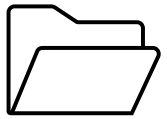
build/<platform>
Temporary build folder used by CMake



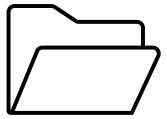
dist/<platform>
Build results/artefacts



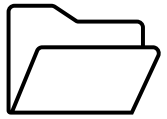
bin
CLI executables and NFIQ2 Library (DLL), shell scripts



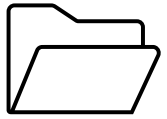
cfg
Configuration files (RF model, training parameters)



data
Artificial sample data



include
NFIQ2 core headers

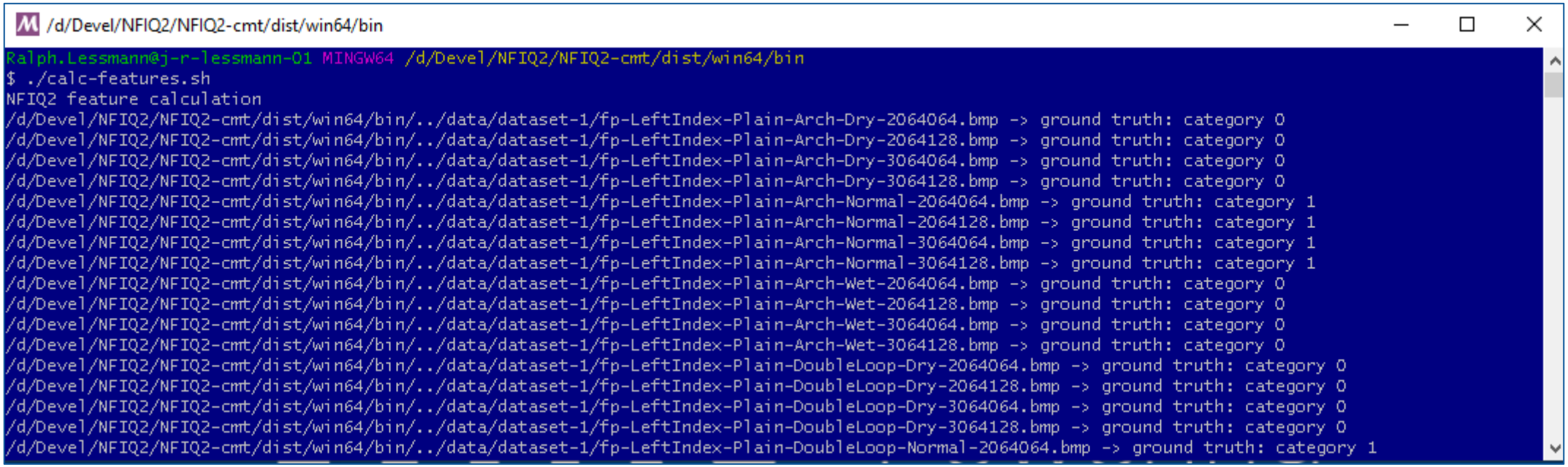


lib
NFIQ2 core libraries

Demonstration

Prepare the probe and training data

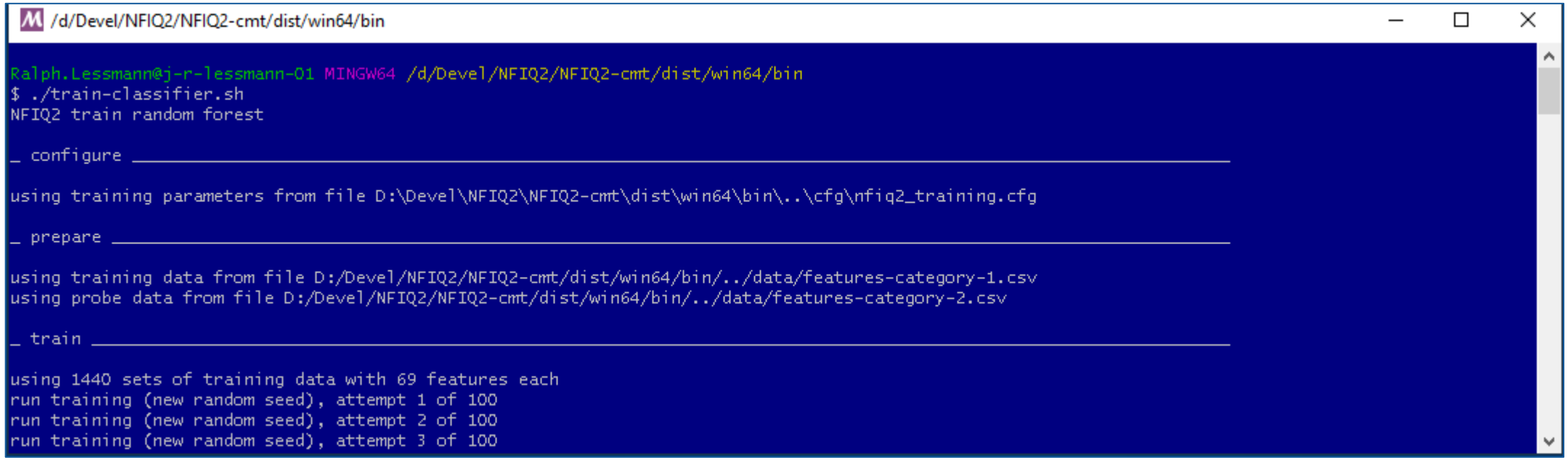
- shell script: *calc-features.sh*
- data: *artificial sample data for training and probe*
- ground truth: *derived from sample data file name (normal:=1, wet and dry:=0)*
- output: *csv files for training and probe*



Demonstration

Perform training

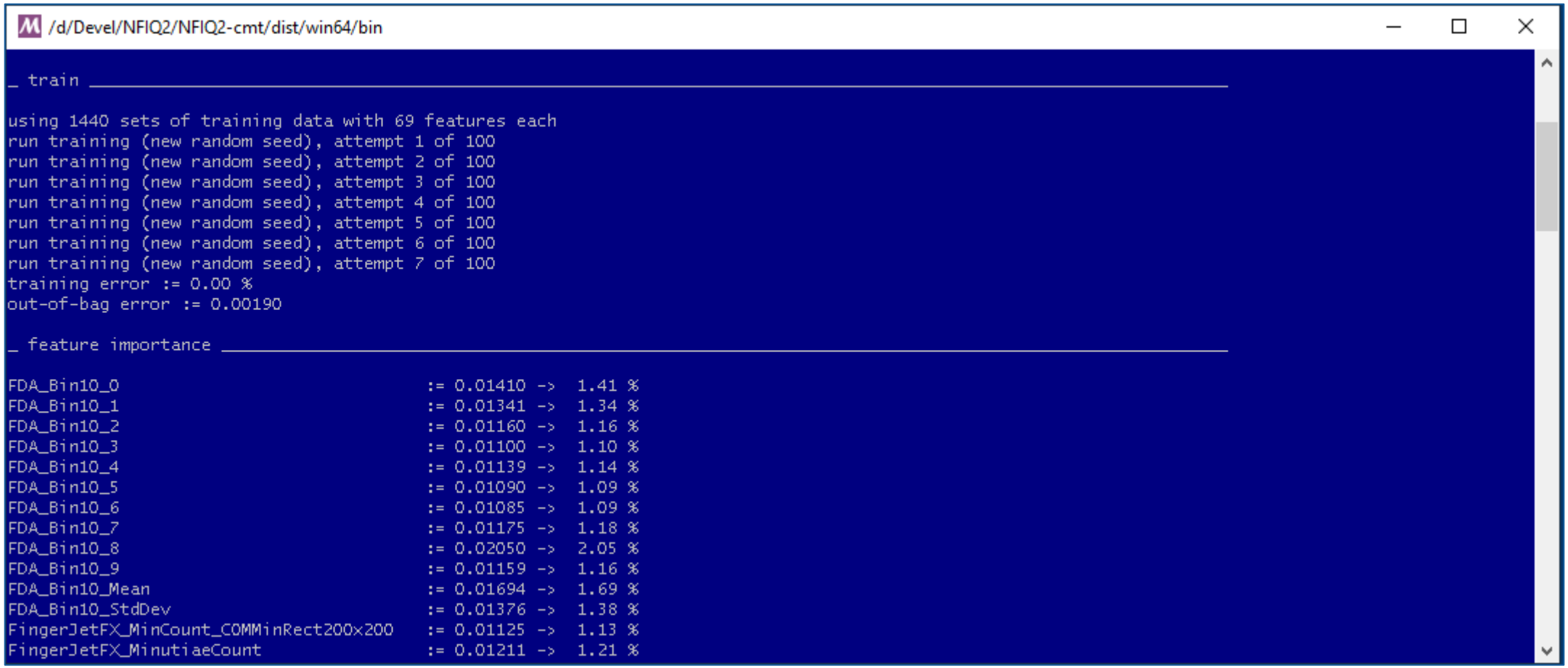
- shell script: *train-classifier.sh*
- data: *csv files for training and probe*
- parameters: *nfiq2_training.cfg*
- output: *training details, convolution matrix, RF model*



Demonstration

Training output

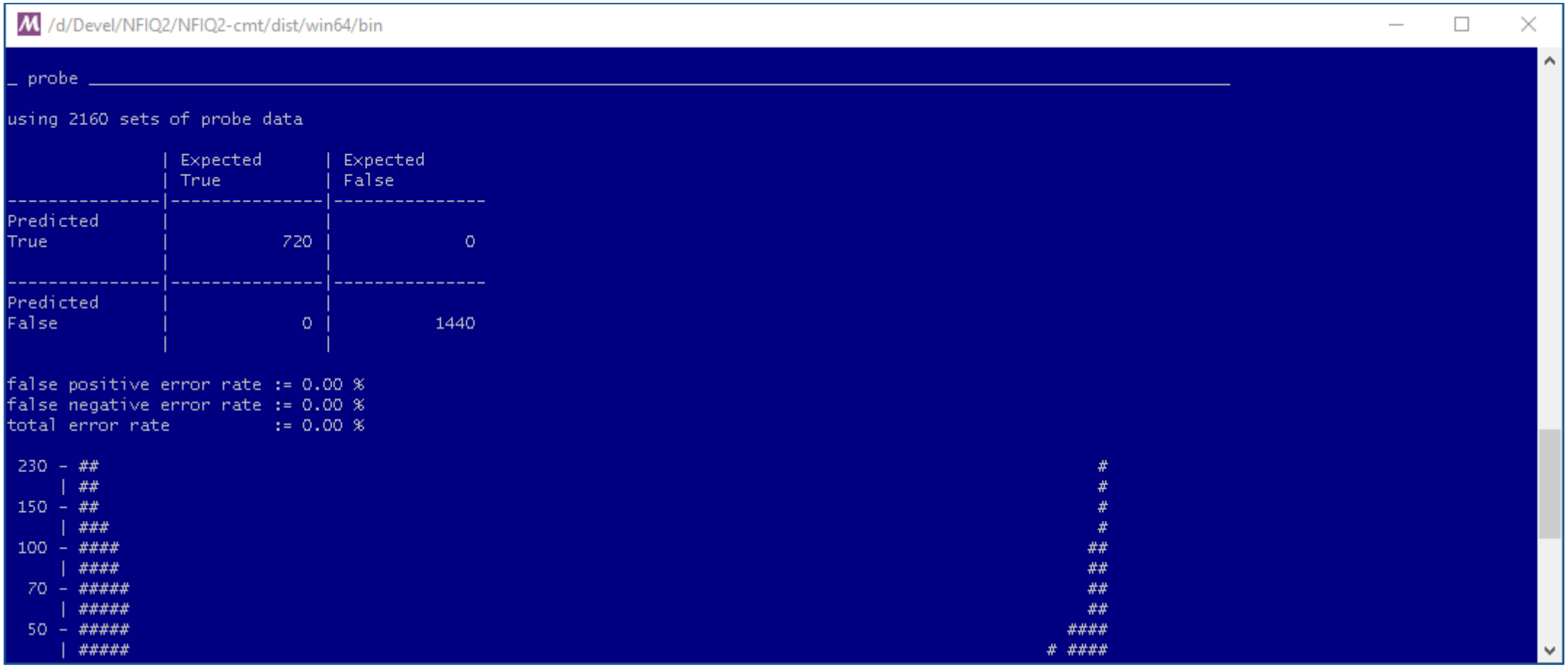
- output: iteration, *training error, out of bag error and feature importance*



Demonstration

Probe output

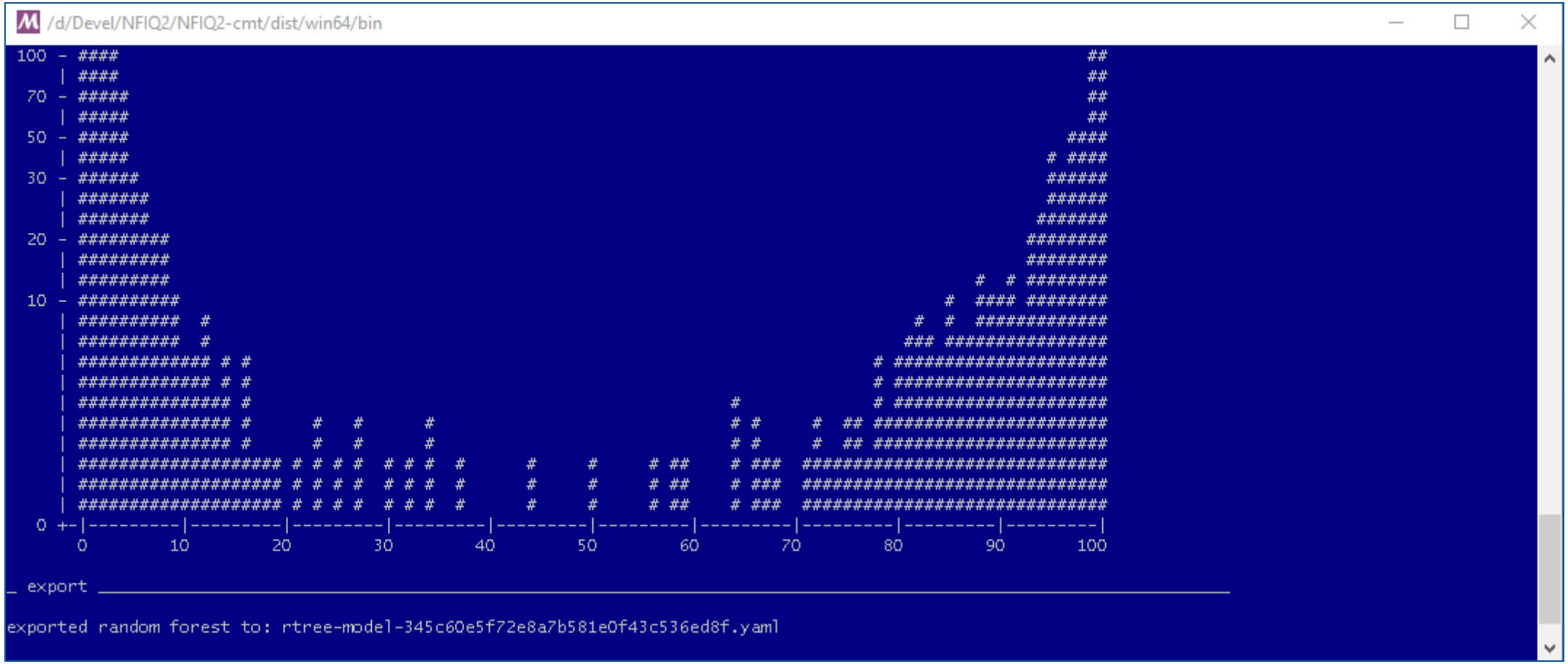
- output: *convolution matrix and error rates*



Demonstration

Probe output

- output: *score histogram, exported RF model (with hash)*



Summary

Achievements

- Common training tools for the NFIQ2 are available
- Re-training the classifier is considered as a task for advanced users
- Re-training shall only be executed if the necessity was proven and justified
- Consider potential interoperability impact before attempting a re-training
- It is recommended to inform ISO/IEC JTC1 SC37 and NIST about any retraining attempt

Next steps

- Consider together with ISO/IEC JTC1 SC37 and NIST on how to make the repository for the training tools available

Out of scope

- Modification of the NFIQ2 feature vector
- Modification of the training tools to fetch the data from databases etc.

Question and Answers

Ralph Lessmann

Director Software Solutions
HID Global

Mobile +49 (0) 172 370 1615
ralph.lessmann@hidglobal.com

Cross Match Technologies GmbH
Unstrutweg 4
D-07743 Jena

Affiliations

European Association for Biometrics
Chair of the Industry Special Interest Group

ISO/IEC JTC1 SC 37 (National Body Germany)
Working Group 3 & 4

Thank you

hidglobal.com