User crossvalidation

Methodology for evaluation of Adaptive Biometric Systems

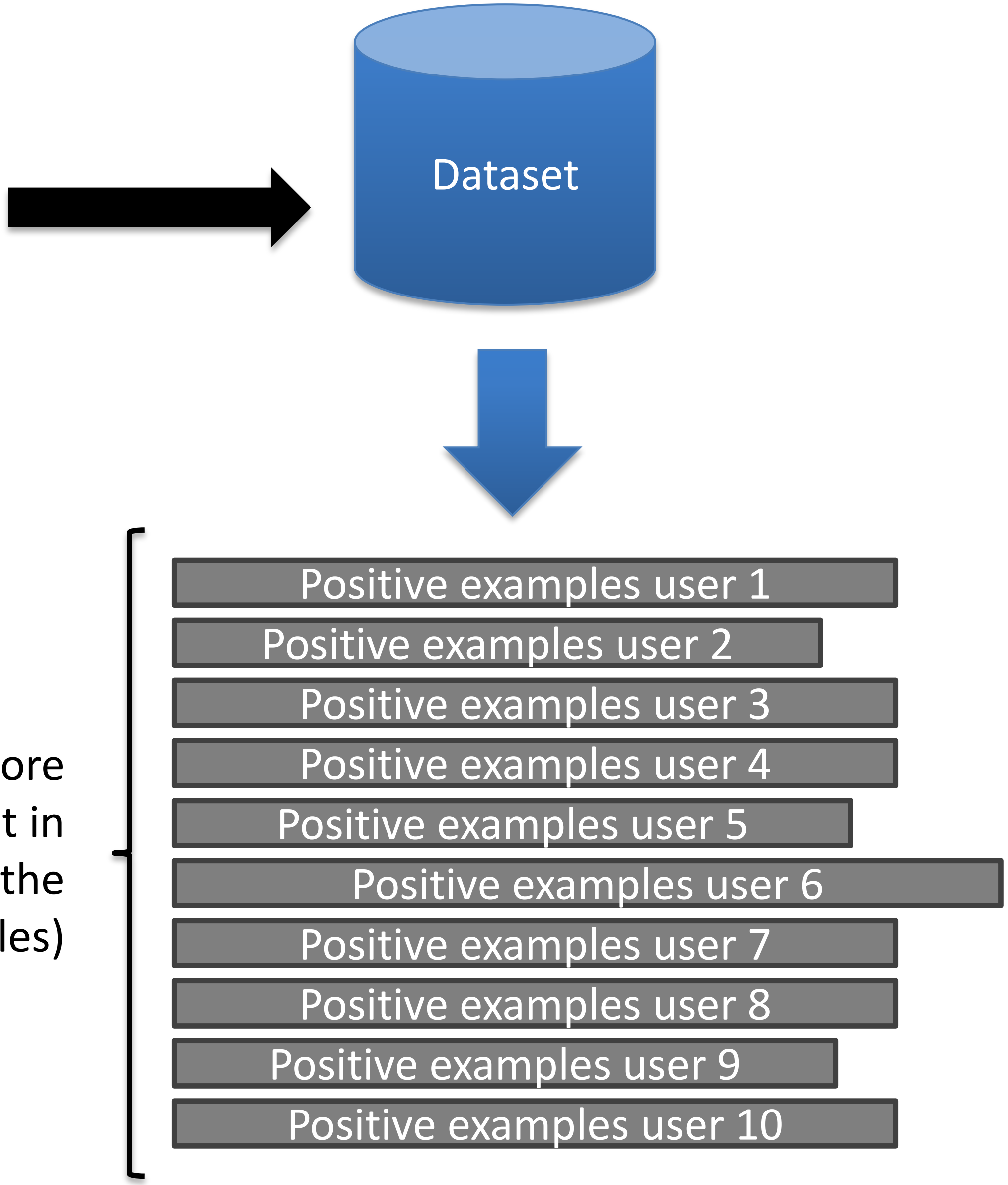
Applied on:

- Pisani, P. H., Lorena, A. C., de Leon Ferreira de Carvalho, A. C. P. Ensemble of Adaptive Algorithms for Keystroke Dynamics. Brazilian Conference on Intelligent Systems 2015.
- Pisani, P. H., Lorena, A. C., de Leon Ferreira de Carvalho, A. C. P. Adaptive Approaches for Keystroke Dynamics. IEEE International Joint Conference on Neural Networks - IJCNN 2015

(and 2 other papers under review)

(Hypothetical example, for a dataset of 10 users)

The original dataset has data from **10 users** (all typed the same expression => same feature set)



Dataset

Positive examples user 1

Positive examples user 2

Positive examples user 3

Positive examples user 4

Positive examples user 5

Positive examples user 6

Positive examples user 7

Positive examples user 8

Positive examples user 9

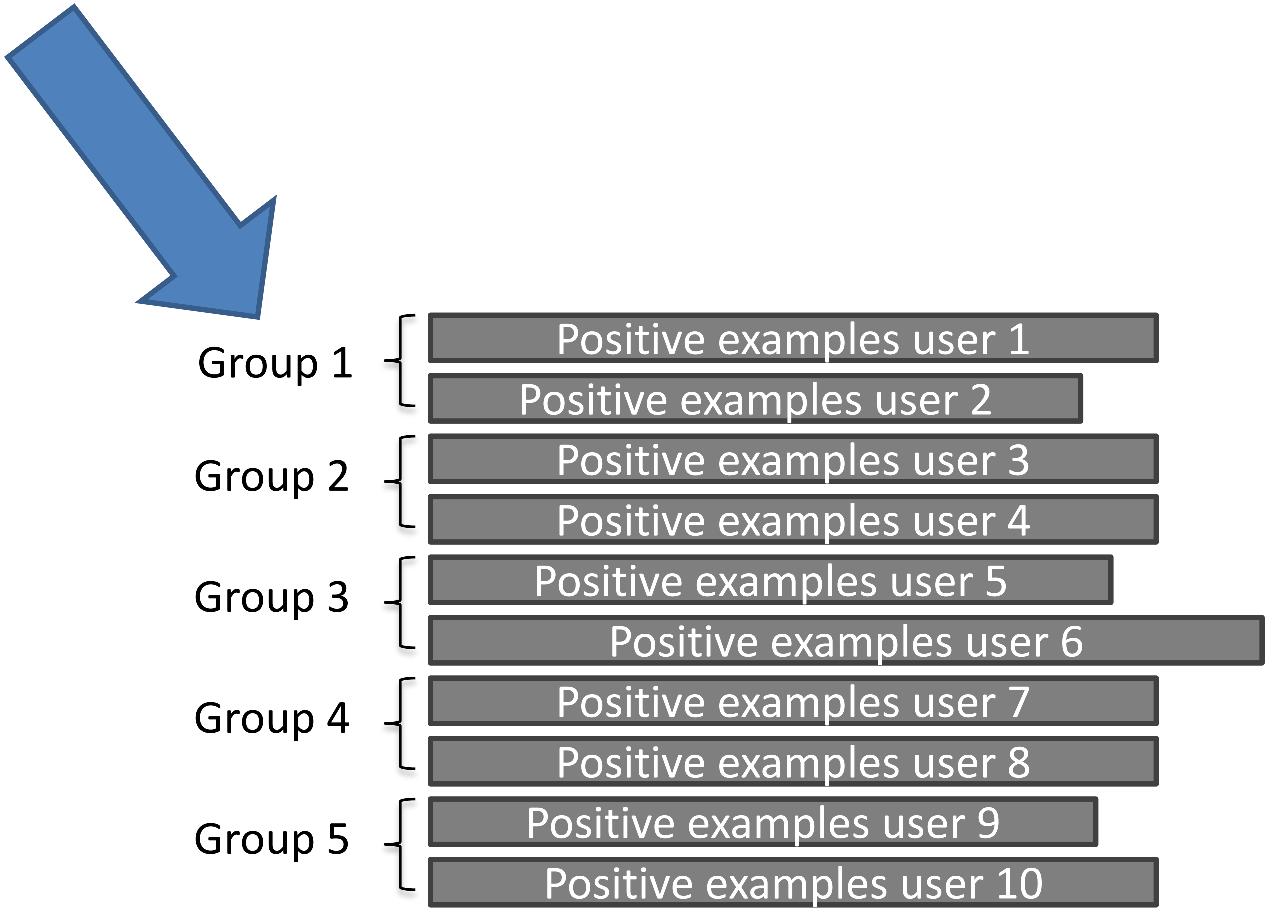
Positive examples user 10

Some users may have more examples than others (except in CMU, where all users have the same amount of examples)

1. Divide users in 5 groups

Each group has aproximately the same size (if the number of users is a multiple of 5, then all groups have the same size).

The division is done randomly, so, in practice, it is very unlikely that the groups are divided the way shown here (sorted).



Group 1
Positive examples user 1
Positive examples user 2

Group 2
Positive examples user 3
Positive examples user 4

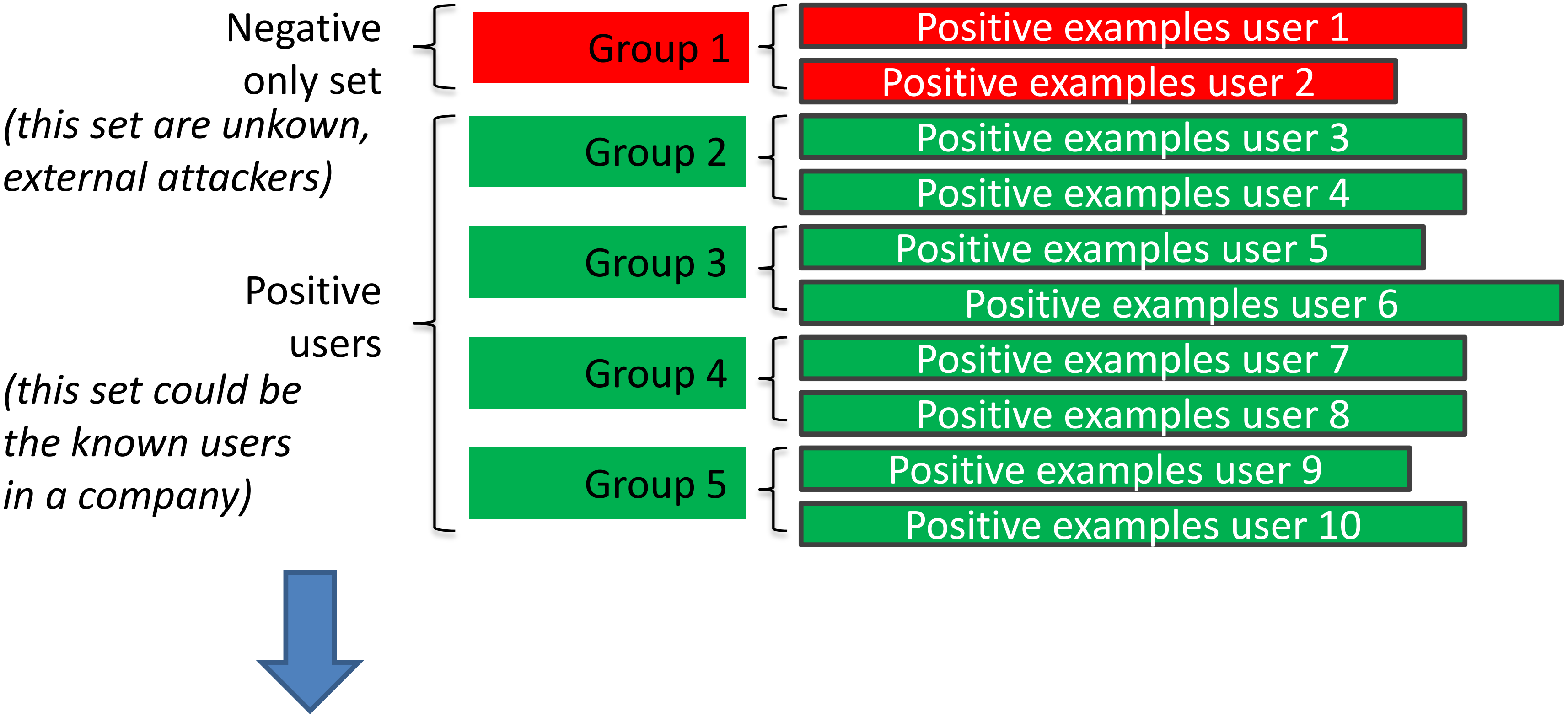
Group 3
Positive examples user 5
Positive examples user 6

Group 4
Positive examples user 7
Positive examples user 8

Group 5
Positive examples user 9
Positive examples user 10

2. Execute experiments

For each experiment, one group is the **negative only set**, and the other ones will be **positive users** **(there are 5 experiments, because all five fold/group combinations are evaluated)**.



Negative only set

(this set are unkown, external attackers)

Positive users

(this set could be the known users in a company)

Group 1

Group 2

Group 3

Group 4

Group 5

Positive examples user 1

Positive examples user 2

Positive examples user 3

Positive examples user 4

Positive examples user 5

Positive examples user 6

Positive examples user 7

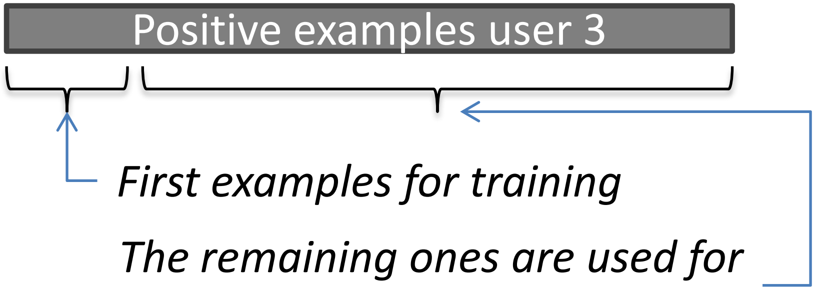
Positive examples user 8

Positive examples user 9

Positive examples user 10

Biometric data streams are generated only for the positive users (in this case, users 3 to 10). The Negative only set are users unknown to the system.

2. Execute experiments



First examples for training

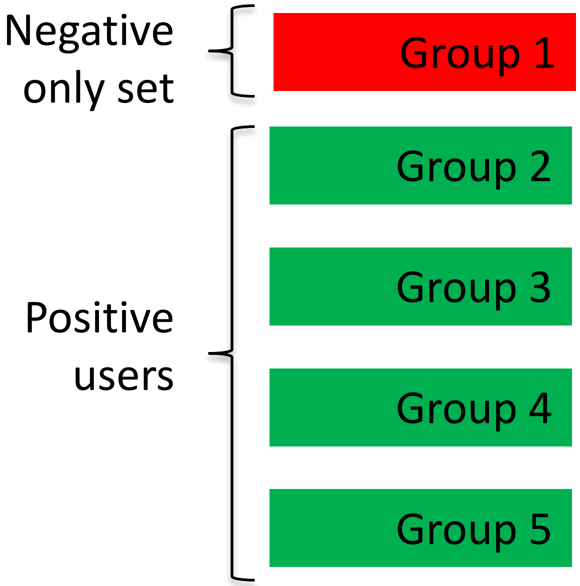
The remaining ones are used for the test data stream.

1. Biometric data streams are generated only for the positive users (in this case, users 3 to 10). The Negative only set are users unknown to the system. **So, there will be 8 data streams**.
2. Firstly, the first 40 examples of the user are employed to induce the model.
3. Then, a test biometric data stream is generated. This test data stream is composed by 70% of positive examples (the remaining examples of the user) and 30% of negative examples. The examples are interleaved randomly.

Note that this defines the length of the stream. If the user has 200 examples in the dataset, 40 are used for training, so 160 can be used for testing.

As these 160 will be 70% of the stream, the test stream length is 160/0.7 = 228. So, there will be 68 negative examples (228 – 160).

2. Execute experiments



4. So, each test stream has 30% of negative examples;
5. These 30% of negative examples can came from the negative only set (external attack) or from other positive users (internal attack).
6. There is a 50% chance of getting a negative example from an external attack and 50% from an internal attack.
7. For user 3, it means that there is a 50% of getting negative examples from users 1 and 2 (negative only) and 50% of getting negative examples from users 4 to 10 (the other positive users).

2. Execute experiments

In the end, there will be:

- 5 experiments (one for each fold/group combination)
- 8 users tested per fold (2 users are left as negative only per fold)
- 30 executions per user (the stream is generated 30 times for each user)

It is called ***user crossvalidation*** because the users are crossvalidated
(not the examples of each user)