CINTI 2020
November 5-7, 2020

Two-factor, continuous authentication framework for multi-site large enterprises

Lehel Dénes-Fazakas ,
Eszter Kail ,
Rita Fleiner

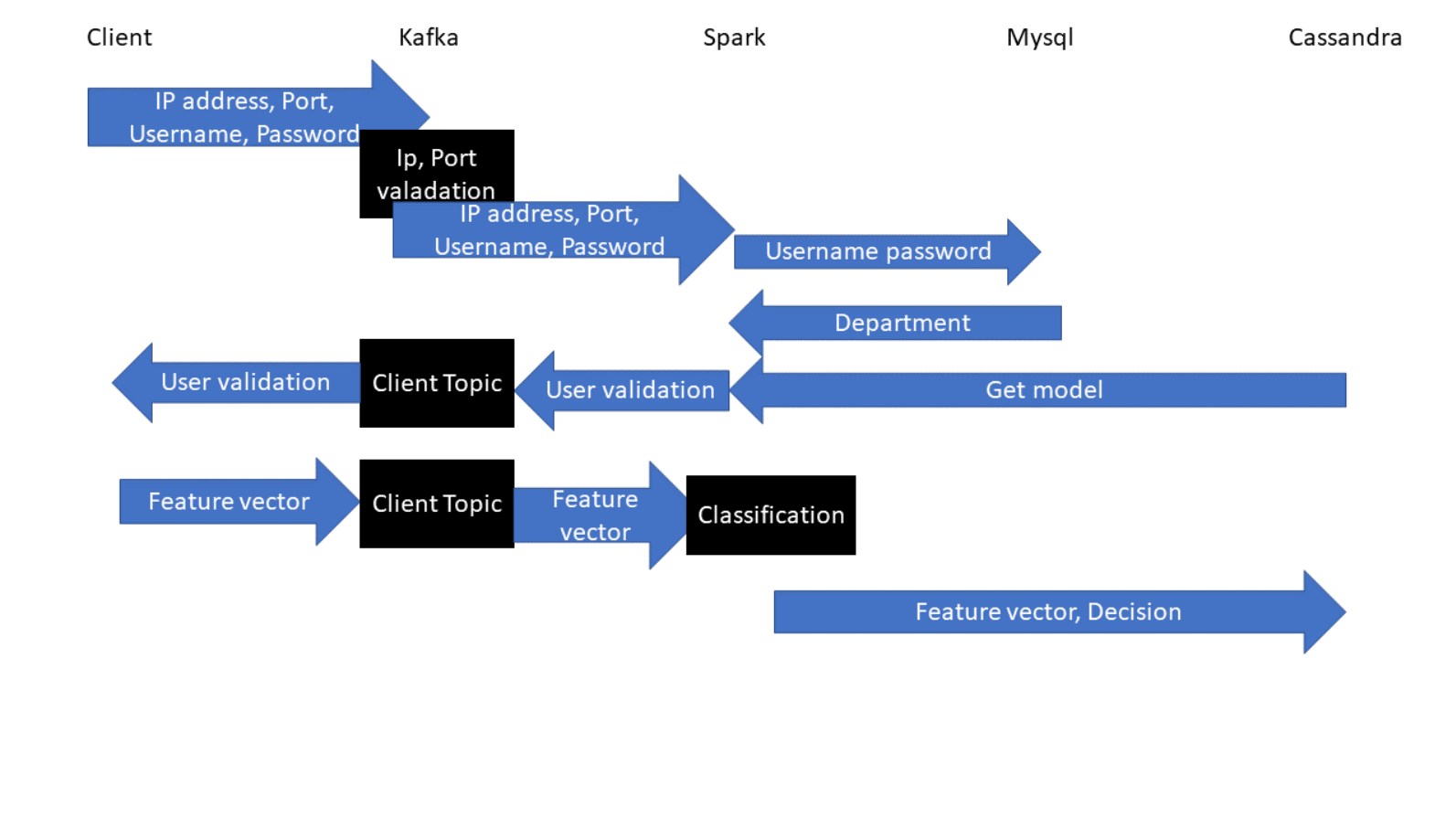
Overview

- Introduction
- Goal – planned system
- Feature extraction
- Create a user model
- Results

Goal

- platform-free 2-factor authentication system for multi location-based companies
- Two factor authentication: at the beginning of a session a user-password pair –, and during the whole session a biometrics based authentication
- Architecture elements:
  - Apache Kafka - For data collection and distribution
  - Apache Spark - cluster-computing framework
  - Apache Cassandra for data storage
  - MySql – for storing the username-pasword authentication details

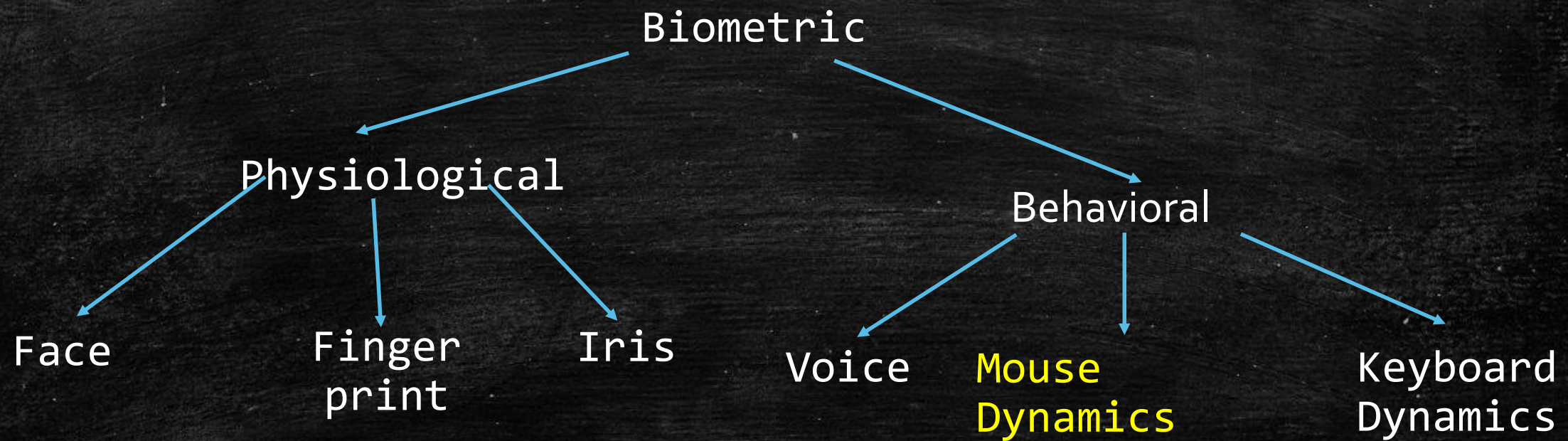
Data-flow for the planned system



Authentication - types

Identity: the process by which the identity of the individual can be clearly determined

We planned to use mouse dynamics based authentication.



Feature extraction

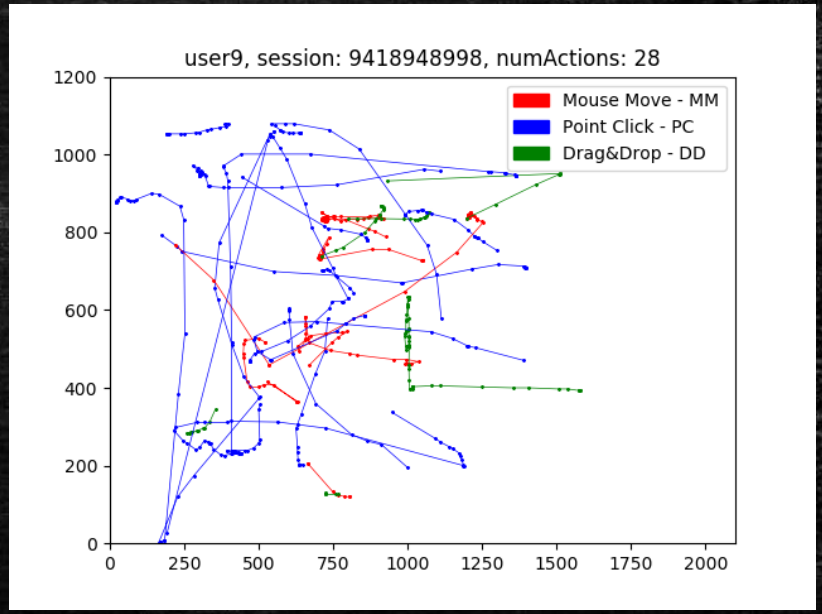
- We have broken down mouse events into 3 different types of mouse actions

Point Click

Drag and Drop

Mouse Move

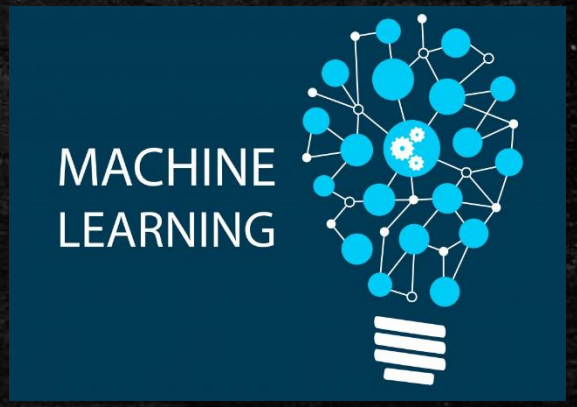
- 23 features extracted from each mouse operation (e.g. speed, acceleration, distance...)



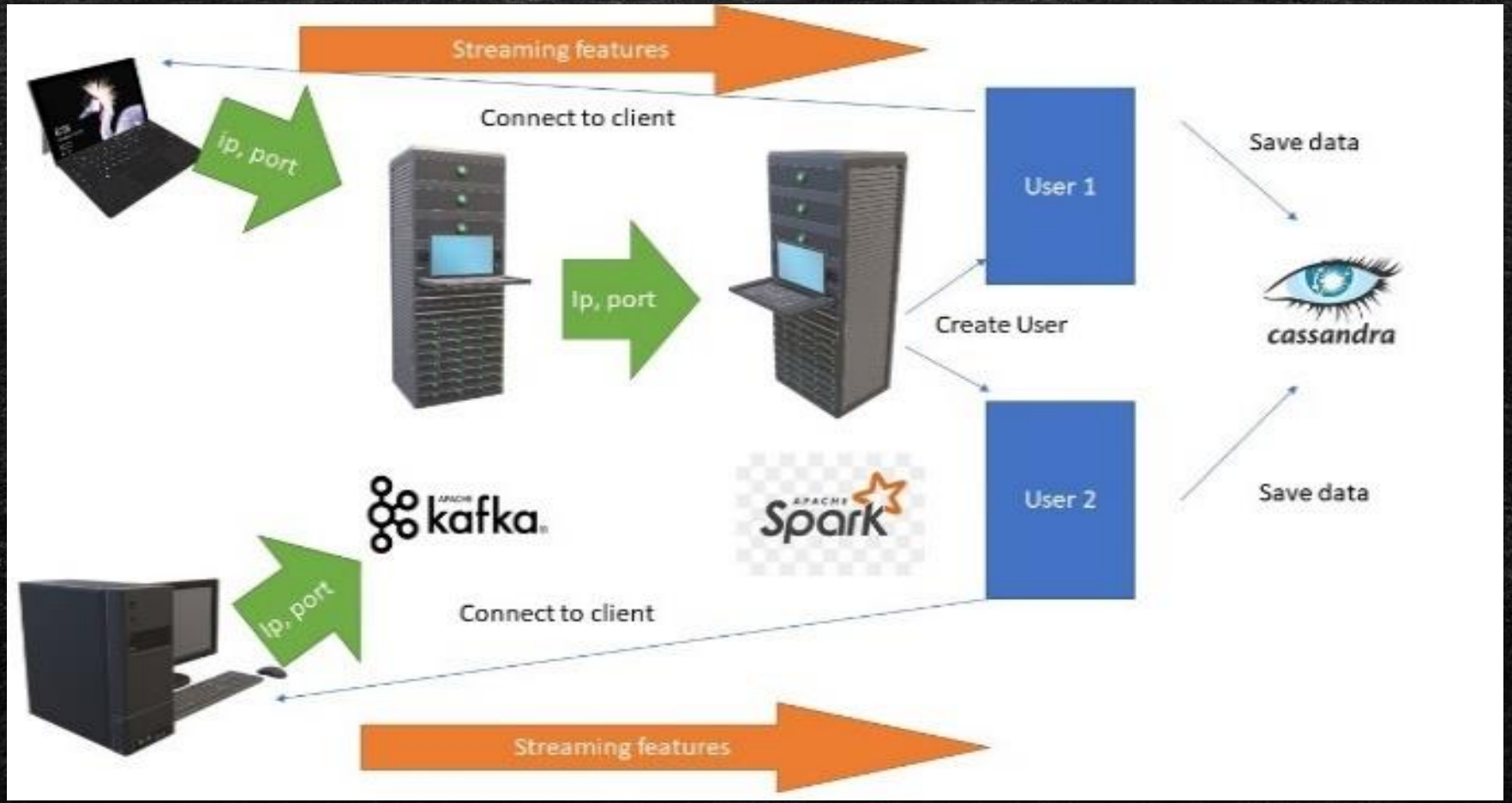
Create a user model

- Binary classifier
  - Random Forest (Max. depth 10, number of trees 100) – we tested several classifiers and based on the perfromance RF was chosen (see Results(2))
- Train data
  - 1000 mouse actions from the user (positive samples)
  - 1000 mouse actions from other users (negative samples)

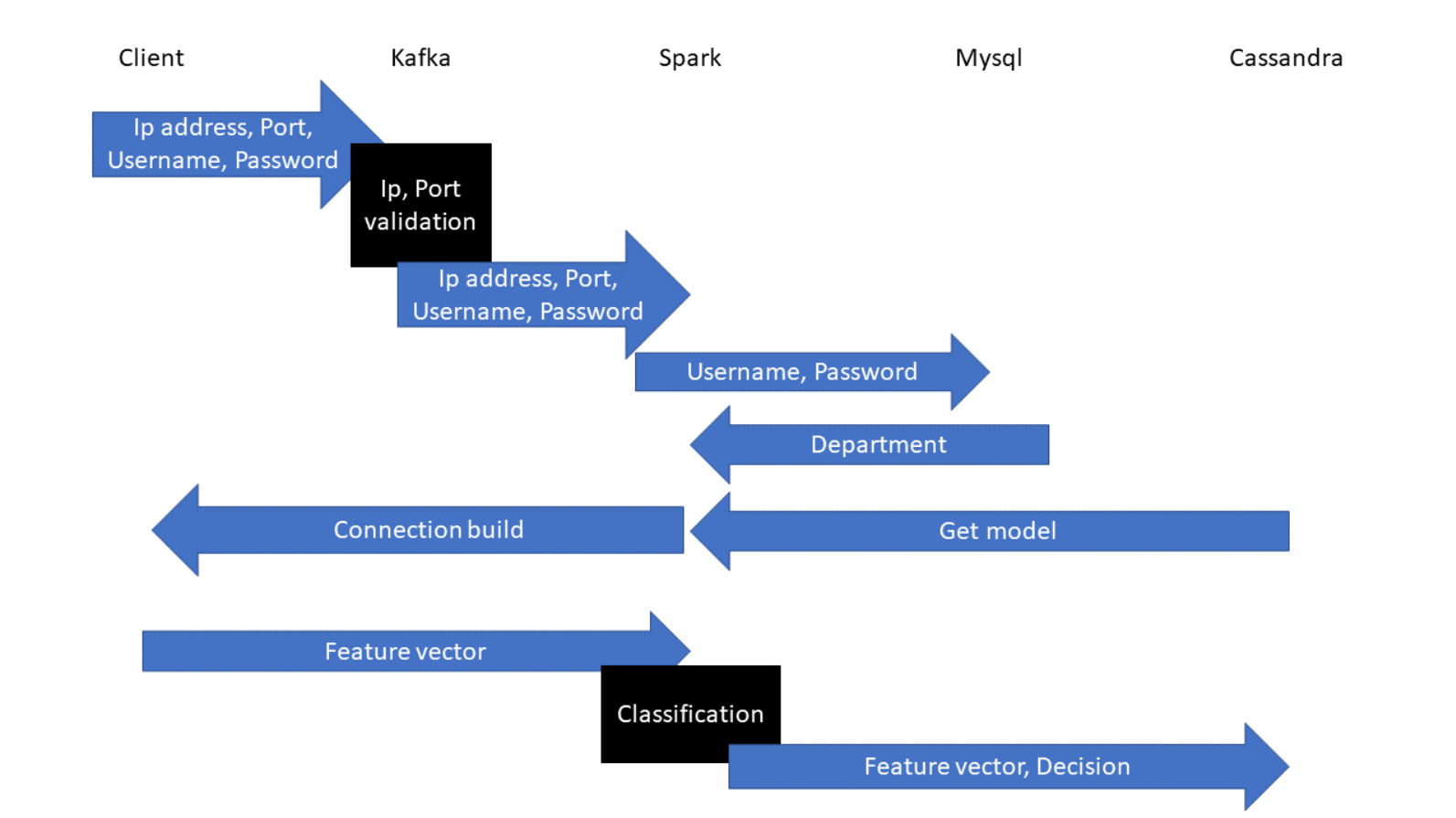
  These values were determined also based on preliminary tests (see Results(1))
- External library used
  - Java Weka 3.8 Machine Learning Library



Architecture of our prototype



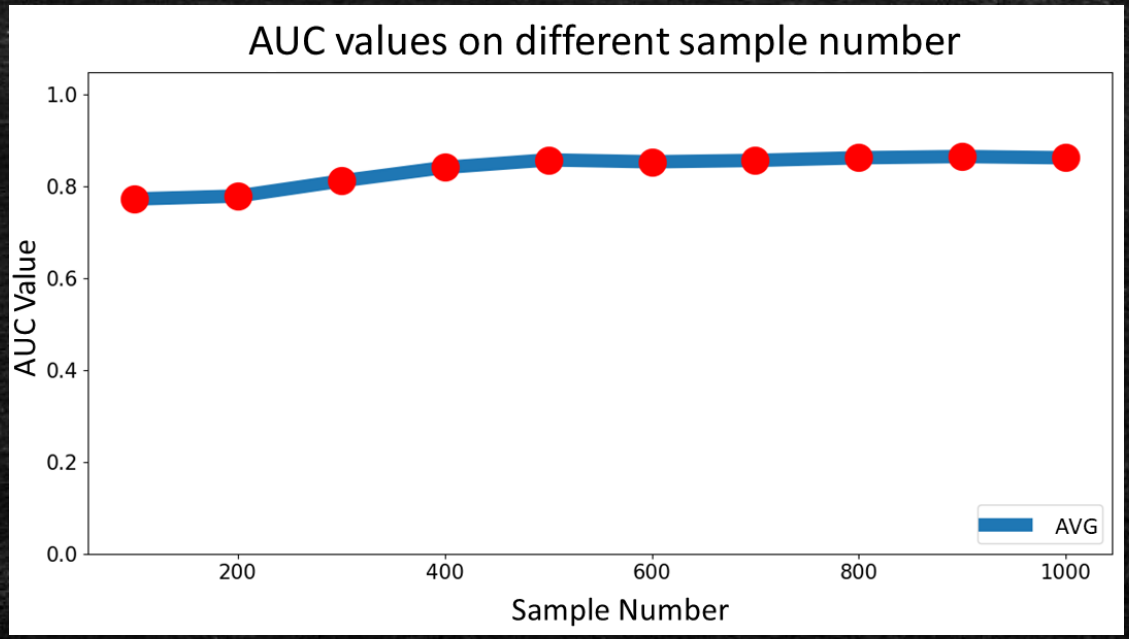
Dataflow of the prototype system



Results(1)

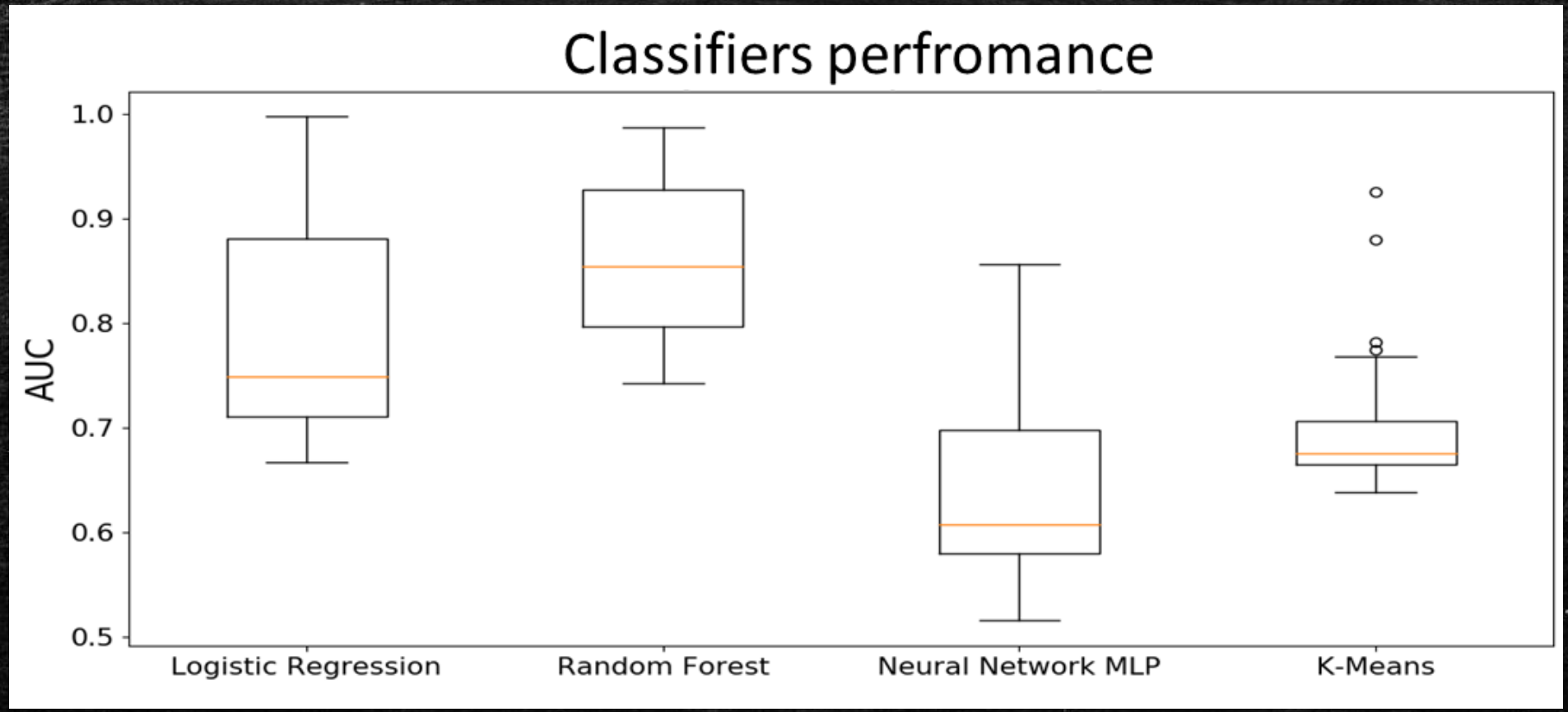
Impact of the number of teaching patterns on the performance of the Classifier (Random Forest)

To determine the number of samples to train the user models



Result(2)

Compare the perfromance of the different classifiers



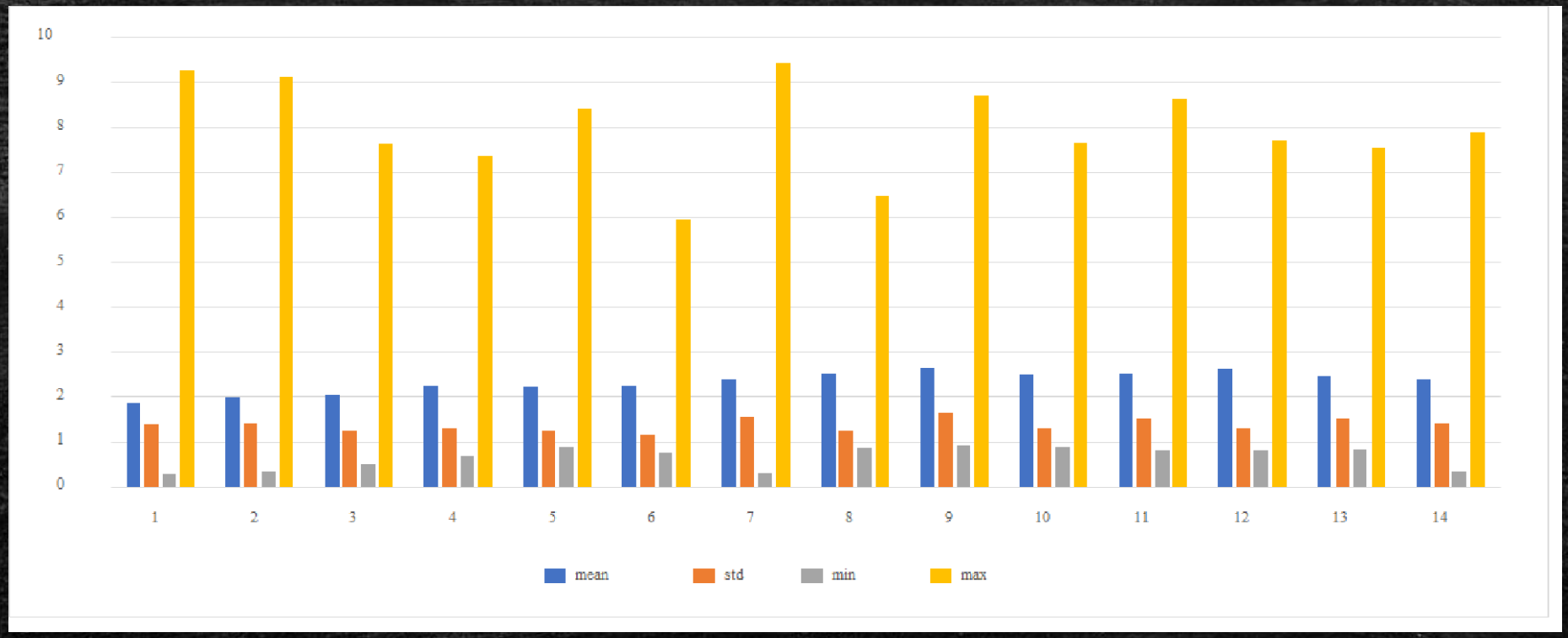
Result(3)

Different read-in types in a scenario where 9 consecutively logged in users were in the system.

	AVG (sec)	STD (sec)	MIN (sec)	MAX (sec)
Cassandra	0.04803	0.02356	0.01855	0.22041
MySQL	1.81578	7.60228	0.4797	72.8310
file	0.012867	0.00602	0.0089	0.05485

Result(4)

Classification time and the effect of the increasing number of users in the system.



14 users logged into the system sequentially with minimal delays. The figure shows that the number of the users in the system does not affect the classification performance.

Result(5)

Classification time and the effect of the increasing number of users in the system.

	mean	std	min	max
1	1.8561	1.3859	0.2723	9.248
2	1.9914	1.4085	0.3296	9.1325
3	2.0541	1.231	0.4997	7.623
4	2.233	1.2871	0.6513	7.35
5	2.2008	1.2402	0.8697	8.4072
6	2.2276	1.1237	0.7561	5.9446
7	2.3638	1.5344	0.3092	9.4325
8	2.4991	1.2421	0.8553	6.4561
9	2.6413	1.628	0.9106	8.6918
10	2.4766	1.3052	0.8792	7.6671
11	2.5098	1.5217	0.7859	8.6201
12	2.6275	1.2942	0.8001	7.6972
13	2.4648	1.5239	0.8369	7.5302
14	2.3677	1.4225	0.3312	7.8661

Summary

- ✓ We have planned a two-factor, continuous authentication system for multi-site enterprises
- ✓ We have implemented a prototype of the system
- ✓ We have conducted feasibility test to prove its usability

Future work:

- Implementing the planned system
- The system components integrating to google cloud platform