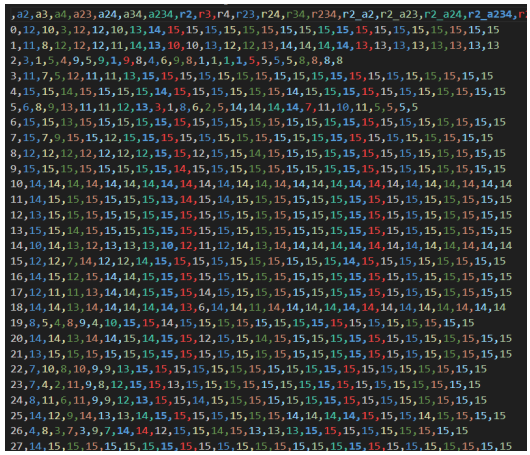
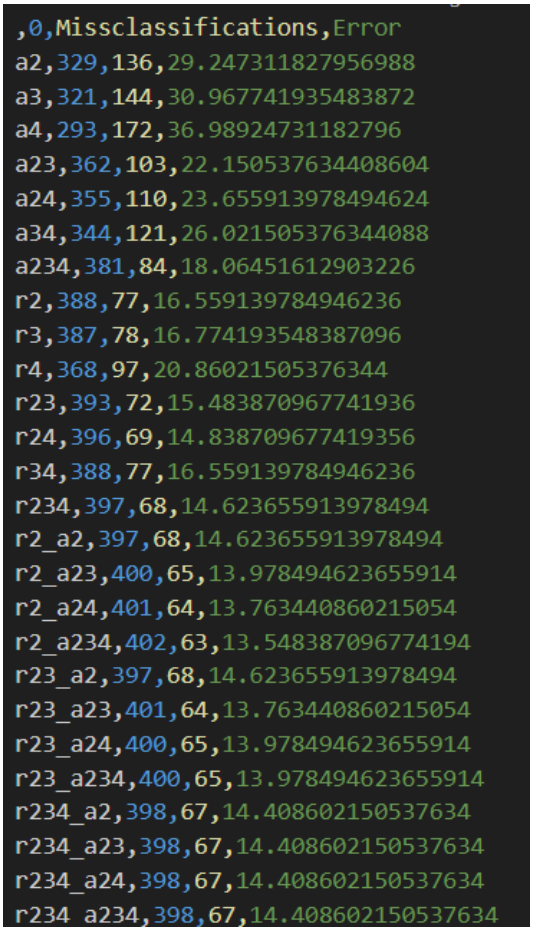
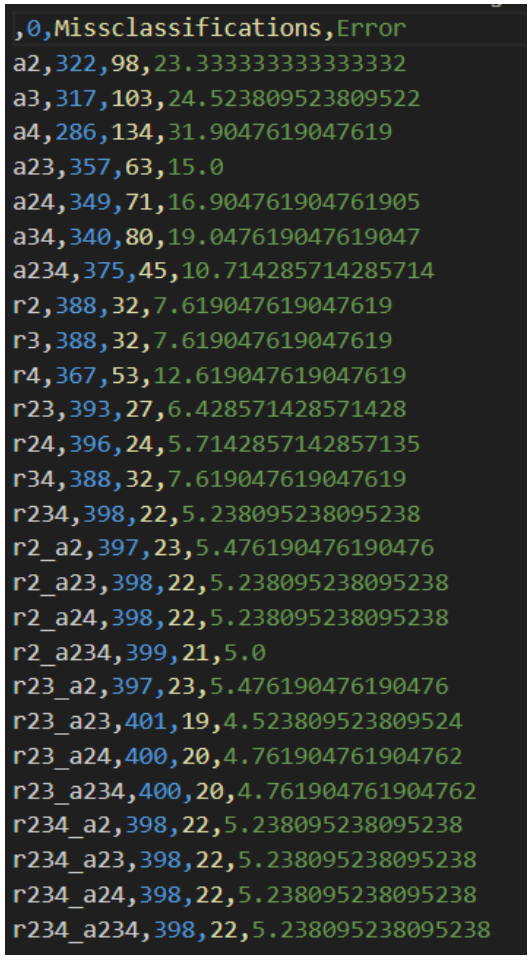
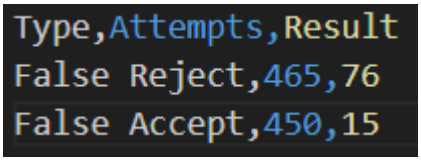
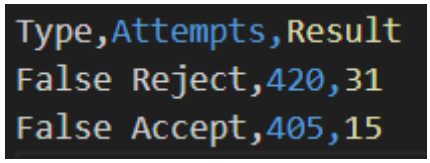
Problem

	Data with 31 User	Data with 28 User
classification		
classification perf		
authentification		

,0,Missclassifications,Error
a2,329,136,29.247311827956988
a3,321,144,30.967741935483872
a4,293,172,36.98924731182796
a23,362,103,22.150537634408604
a24,355,110,23.655913978494624
a34,344,121,26.021505376344088
a234,381,84,18.06451612903226
r2,388,77,16.559139784946236
r3,387,78,16.774193548387096
r4,368,97,20.86021505376344
r23,393,72,15.483870967741936
r24,396,69,14.838709677419356
r34,388,77,16.559139784946236
r234,397,68,14.623655913978494
r2_a2,397,68,14.623655913978494
r2_a23,400,65,13.978494623655914
r2_a24,401,64,13.763440860215054
r2_a234,402,63,13.548387096774194
r23_a2,397,68,14.623655913978494
r23_a23,401,64,13.763440860215054
r23_a24,400,65,13.978494623655914
r23_a234,400,65,13.978494623655914
r234_a2,398,67,14.408602150537634
r234_a23,398,67,14.408602150537634
r234_a24,398,67,14.408602150537634
r234_a234,398,67,14.408602150537634

,0,Missclassifications,Error
a2,322,98,23.333333333333332
a3,317,103,24.523809523809522
a4,286,134,31.9047619047619
a23,357,63,15.0
a24,349,71,16.904761904761905
a34,340,80,19.047619047619047
a234,375,45,10.714285714285714
r2,388,32,7.619047619047619
r3,388,32,7.619047619047619
r4,367,53,12.619047619047619
r23,393,27,6.428571428571428
r24,396,24,5.7142857142857135
r34,388,32,7.619047619047619
r234,398,22,5.238095238095238
r2_a2,397,23,5.476190476190476
r2_a23,398,22,5.238095238095238
r2_a24,398,22,5.238095238095238
r2_a234,399,21,5.0
r23_a2,397,23,5.476190476190476
r23_a23,401,19,4.523809523809524
r23_a24,400,20,4.761904761904762
r23_a234,400,20,4.761904761904762
r234_a2,398,22,5.238095238095238
r234_a23,398,22,5.238095238095238
r234_a24,398,22,5.238095238095238
r234_a234,398,22,5.238095238095238

Type,Attempts,Result
False Reject,465,76
False Accept,450,15

Type,Attempts,Result
False Reject,420,31
False Accept,405,15

Notes

- classification performance is better for r2, r3, r4 (and the combinations based on them)
- the classification performance for a2, a3, a4 is also better
- (r2, r3, r4) and (a2, a3, a4) are independant

User Data

Digraphs (2-graphs)

user	means_mean	meadians_means	stds_mean	
0	476	403	193	
1	310	289	74	
2	1596	1147	1278	
3	349	315	138	
4	328	303	102	
5	630	564	205	
6	368	339	120	
7	326	287	141	
8	674	577	221	
9	567	519	162	
10	232	212	62	
11	1133	980	401	
12	960	833	394	
13	717	641	234	?
14	584	538	191	
15	345	325	78	
16	534	491	156	
17	307	275	108	
18	297	282	61	?
19	336	291	119	
20	481	461	81	
21	552	496	182	
22	523	473	162	
23	860	785	249	
24	407	372	118	
25	505	453	170	
26	578	535	200	?
27	504	455	150	
28	930	822	375	
29	298	282	72	
30	844	762	264	

Trigraphs (3-graphs)

```
user,means_mean,meadians_means,stds_mean
0,820,774,150
1,534,512,82
2,2678,2185,1353
3,682,637,171
4,602,568,124
5,1122,1084,138
6,599,568,126
7,531,501,122
8,827,790,144
9,814,773,157
10,410,392,63
11,1299,1225,287
12,1490,1398,322
13,1160,1113,181
14,927,871,203
15,589,564,91
16,802,762,167
17,532,505,103
18,526,512,58
19,573,547,114
20,693,667,97
21,788,741,168
22,689,646,128
23,1114,1044,233
24,621,595,109
25,786,736,174
26,907,865,147
27,702,664,138
28,1319,1236,279
29,503,485,77
30,1144,1052,300
```

Fourgraphs (4-graphs)

```
user,means_mean,meadians_means,stds_mean
0,1091,1061,116
1,752,739,64
2,3594,3271,976
3,979,943,134
4,854,834,93
5,1510,1491,97
6,838,817,102
7,741,723,83
8,983,952,115
9,1090,1058,132
10,580,561,57
11,1599,1532,223
12,1853,1797,212
13,1560,1538,122
14,1225,1183,156
15,810,791,74
16,1140,1106,146
17,736,722,64
18,743,738,38
19,811,794,77
20,957,940,89
21,1041,1012,115
22,875,864,78
23,1428,1385,180
24,825,811,77
25,1085,1052,138
26,1228,1199,119
27,942,920,92
28,1625,1587,183
29,715,702,56
30,1460,1399,203
```

?

?

?

r-distance

a-distance

- a-distance implementation: subtraction => division
- classification implementation: mean distance

Sending Keystrokes over network

Design

- using websockets:
    - *connection in user space (not managed by browser)*
    - *reliable (no UDP => using Keystrokes is unacceptable)*
    - *minimal parsing overhead (no HTTP header etc.)*
    - *browser build in*

Problems Browser

- when to open network socket (loading page without logging in)
- how to match incoming keystroke stream to user, when user is logging in right now(no valid session)
    - *generate private uuid for each page load, that gets send with keystrokes and login data*
    - *send login data over websocket connection*
- network/async browser | keystrokes don't have to arrive in right order
    - *counter for each message*

Problems Analysis

- variable network latency
    - *error margin based on ttl* (better not)
    - *measure round trip time from server to client to server at connection start*
    - estimated performance similar to browser with low timestamp modification
    - any mitigation strategy can most likely be abused

Different Thought

Free Text Auth

- save classifikation on authentification
    - saves computation
    - not really nessary (without)