User Authentication through Keystroke Dynamics

Johannes Luig

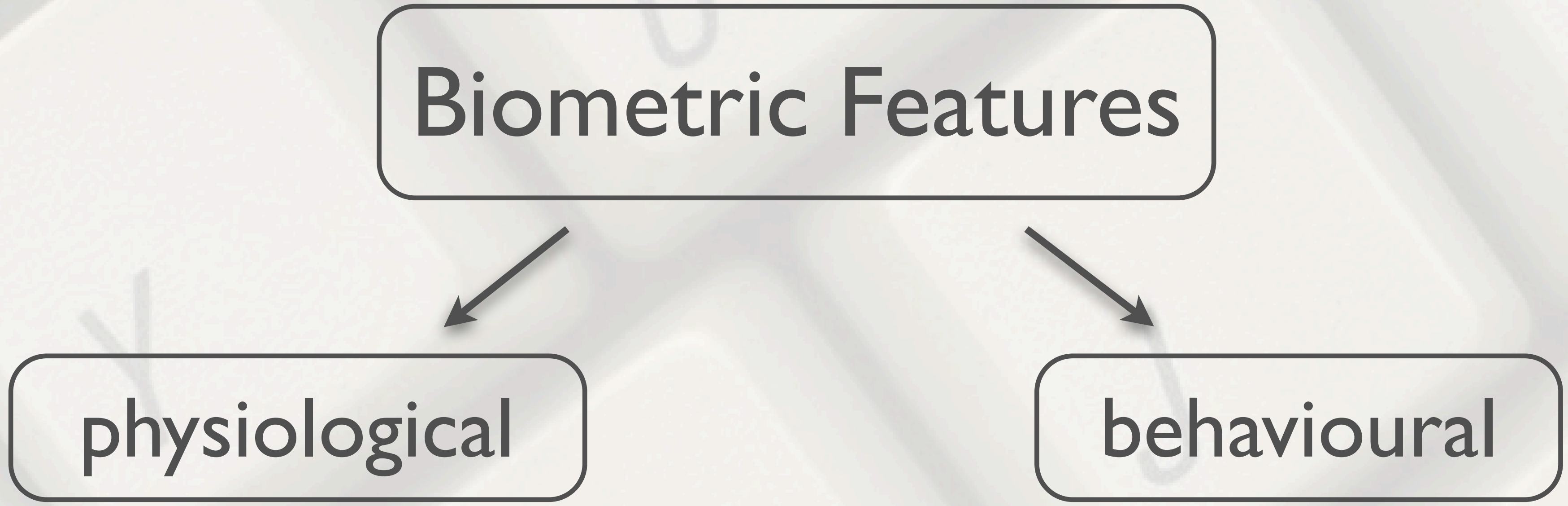
Overview

- Introduction
- Measurable Characteristics
- How to measure „Similarity"?
- User Authentication
- Performance

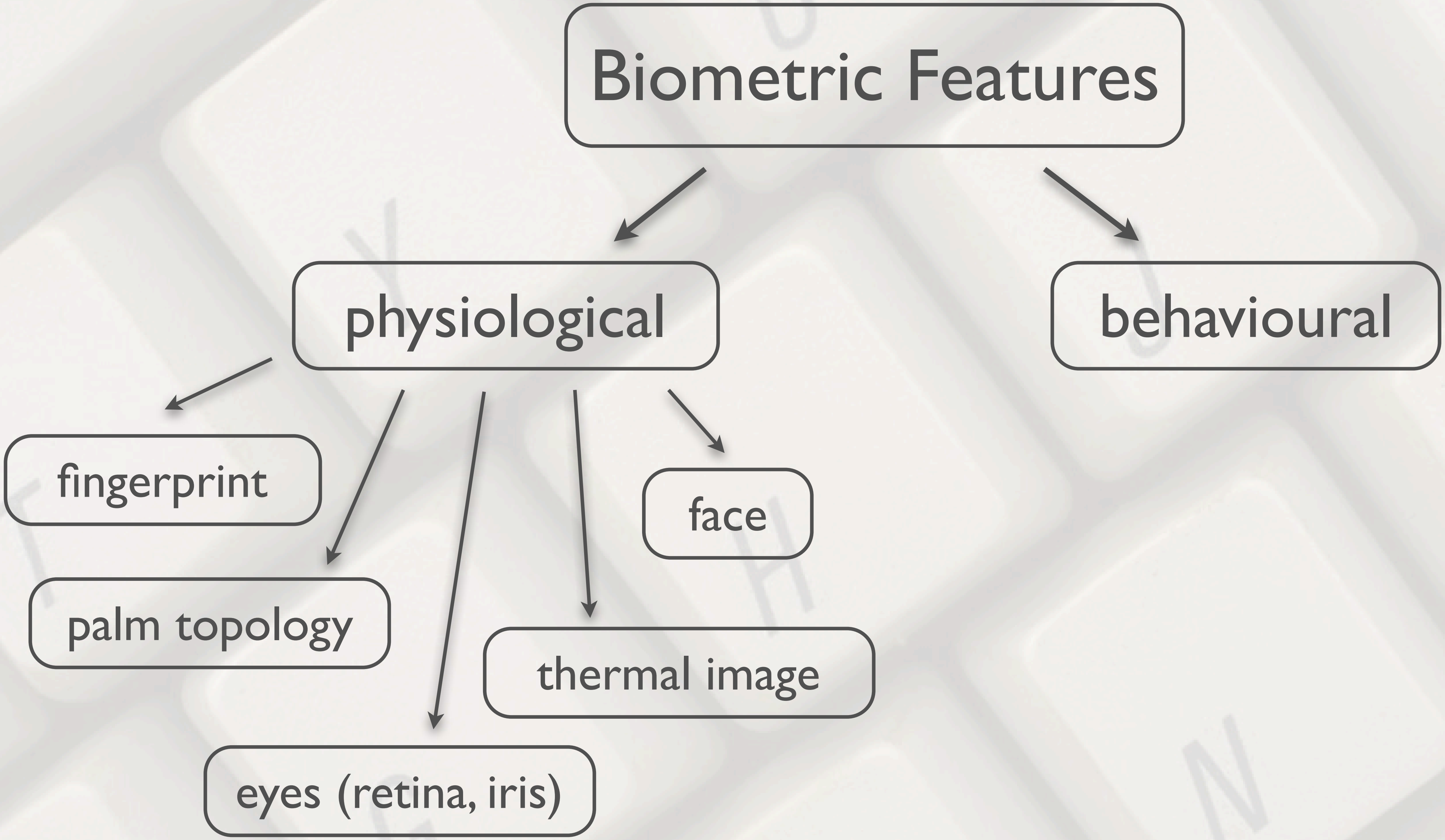
Introduction

Biometric Features

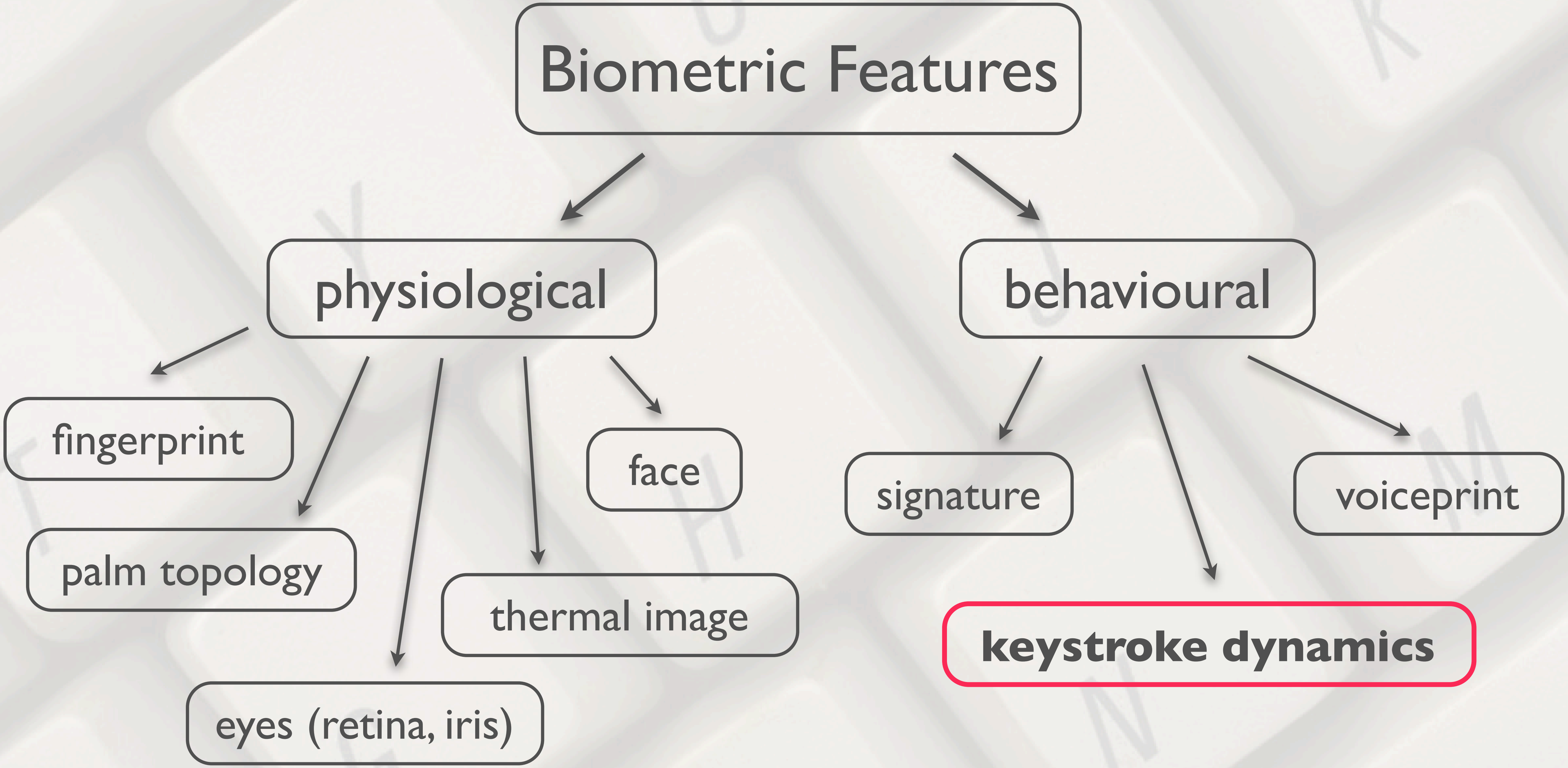
Introduction



Introduction



Introduction



Introduction

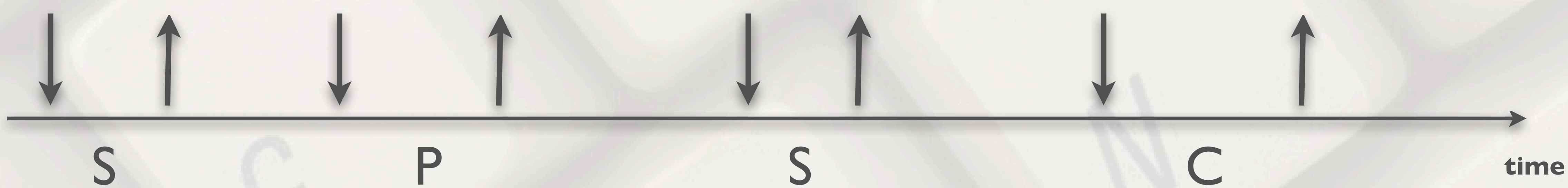
- Motivation:
  - Typing on keyboard does not produce continuous stream of non-stop data, but distinctive patterns
  - Human actions are predictible in the performance of repetitive and routine tasks
  - No specific (expensive) tools needed
  - Method is reasonable (even „hidden check“ possible)

Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency

Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency

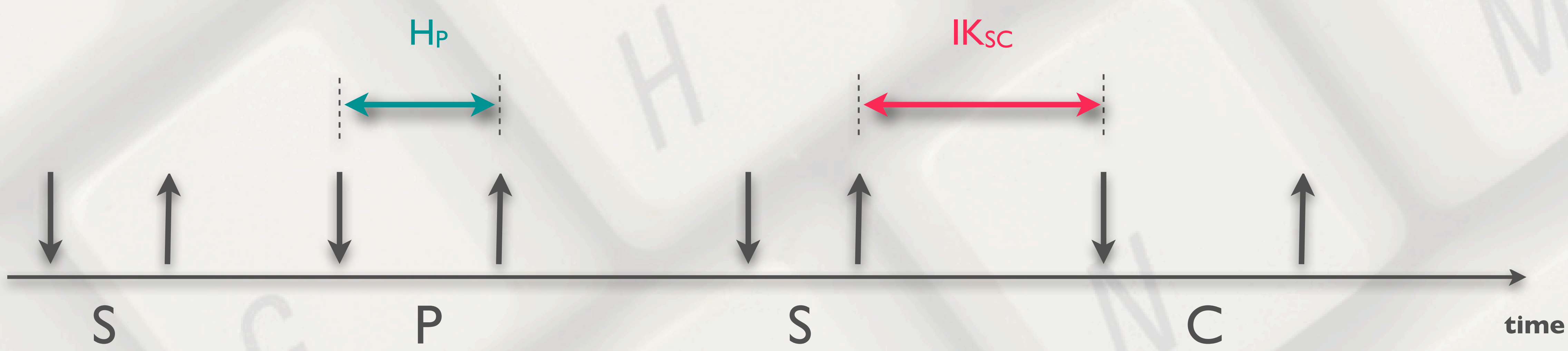


Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency

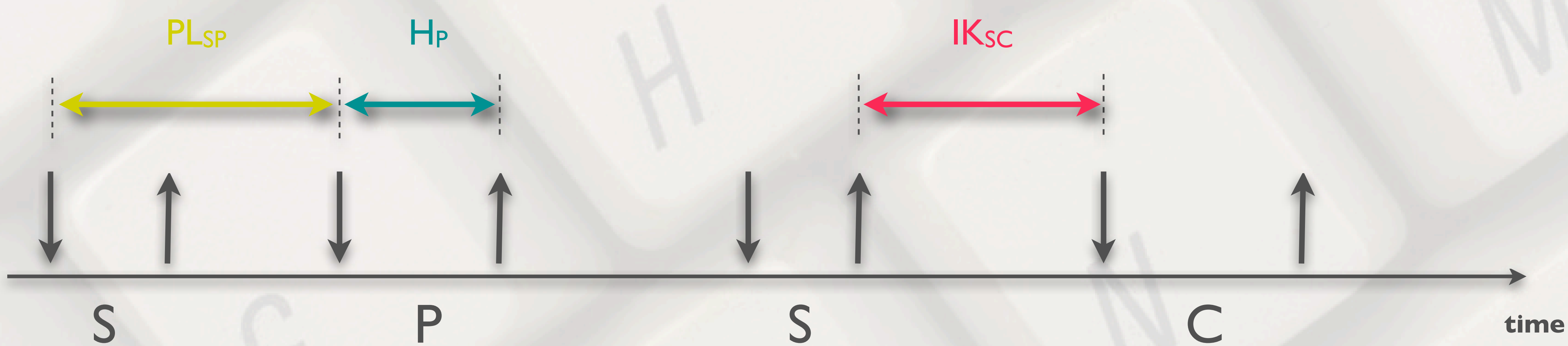
Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency



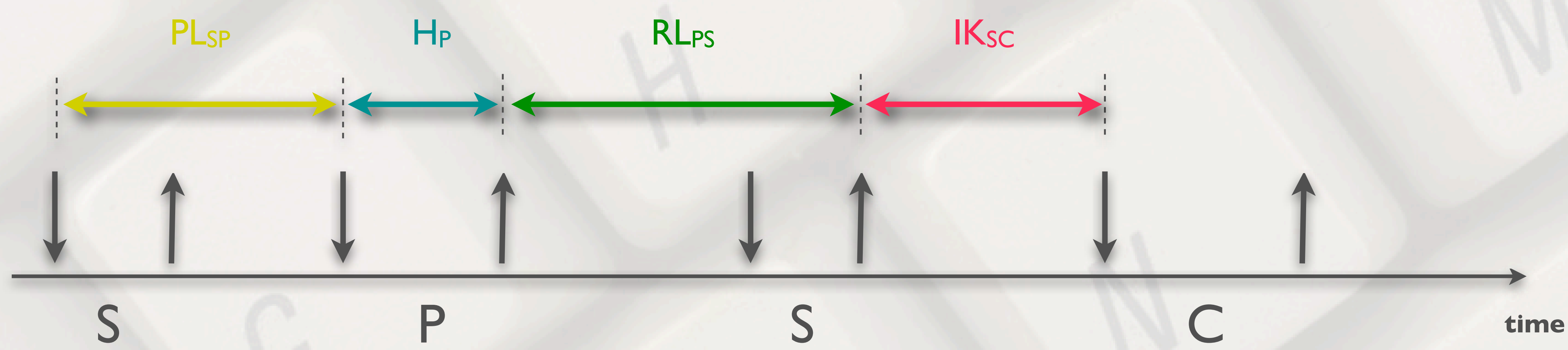
Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency



Measurable Characteristics

- Hold Time
- Interkey Time
- Press/Release Latency

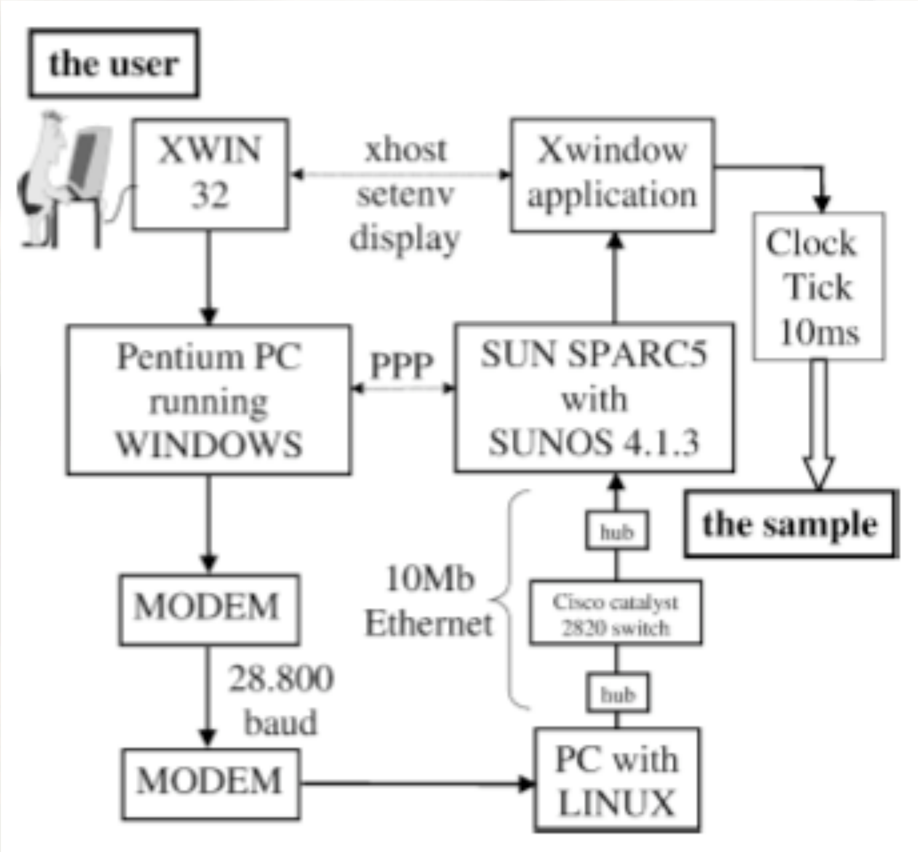


Measurable Characteristics

- Timing
  - Internal CPU clock
    - 8253 timer in IBM-compatible computers
    - BIOS microsecond timing function

Measurable Characteristics

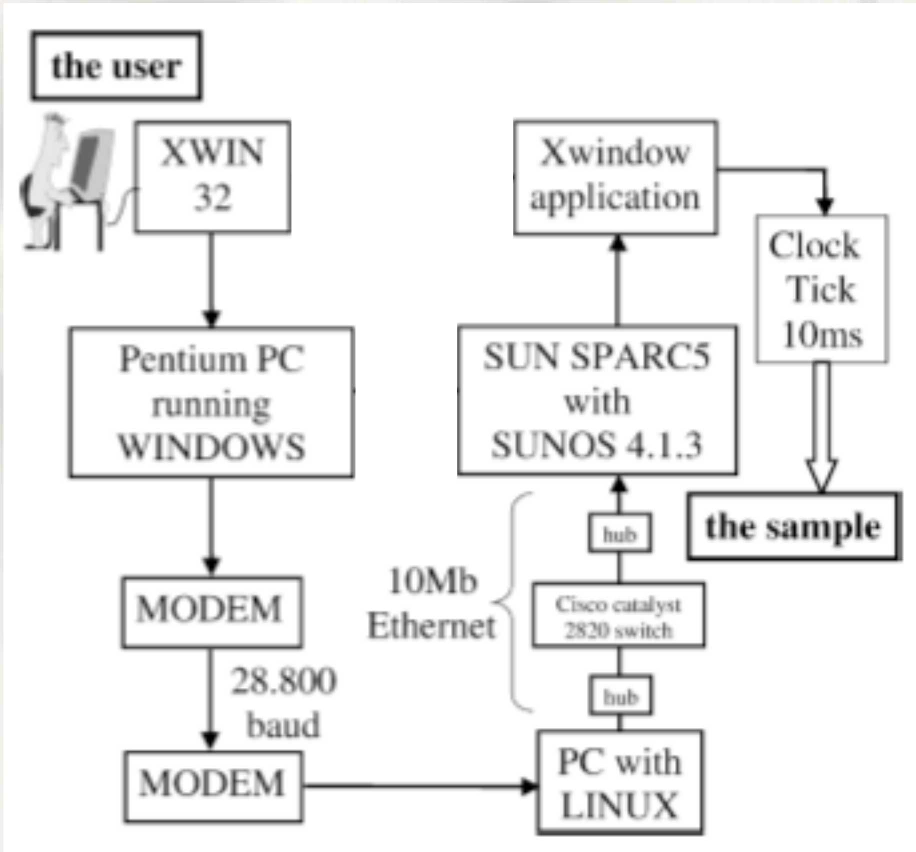
- Timing
  - Example: simulation of remote situation



from Bergadano et al.:
„User Authentication through Keystroke Dynamics“

Measurable Characteristics

- Timing
  - <u>Example:</u> simulation of remote situation



from Bergadano et al.:
„User Authentication through Keystroke Dynamics“

How to measure „Similarity“? (I)

- General approach:
  - Measure hold and/or interkey times
  - Measurement data = vectors in vector space
  - Identify typing person using traditional pattern recognition techniques or Neural Network paradigms

How to measure „Similarity“? (I)

- Pattern recognition techniques include:
  - **k-Means Clustering**
  - Cosine Measure
  - Minimum Distance
  - Bayes‘ Rule
  - Potential Function

How to measure „Similarity“? (I)

- Pattern recognition techniques include:
  - **k-Means Clustering**
  - Cosine Measure
  - Minimum Distance
  - Bayes‘ Rule
  - Potential Function

- cluster data into k partitions
- try to find centers of „natural“ clusters
- minimize intra-cluster variance (squared error):

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

How to measure „Similarity“? (1)

- Pattern recognition techniques include:
  - k-Means Clustering
  - **Cosine Measure**
  - Minimum Distance
  - Bayes‘ Rule
  - Potential Function

How to measure „Similarity“? (I)

- Pattern recognition techniques include:
  - k-Means Clustering
  - **Cosine Measure**
  - Minimum Distance
  - Bayes‘ Rule
  - Potential Function

- cosine of angle between two feature vectors:

$$s^{(C)}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^{\dagger}\mathbf{x}_b}{\|\mathbf{x}_a\|_2 \cdot \|\mathbf{x}_b\|_2}$$

How to measure „Similarity“? (I)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - **Minimum Distance**
  - Bayes‘ Rule
  - Potential Function

How to measure „Similarity“? (1)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - **Minimum Distance**
  - Bayes‘ Rule
  - Potential Function

- euclidean distance:

$$\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

How to measure „Similarity“? (1)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - Minimum Distance
  - **Bayes‘ Rule**
  - Potential Function

How to measure „Similarity“? (1)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - Minimum Distance
  - **Bayes‘ Rule**
  - Potential Function

- relates conditional and marginal probability distributions of random variables to each other

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

How to measure „Similarity“? (I)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - Minimum Distance
  - Bayes‘ Rule
  - **Potential Function**

How to measure „Similarity“? (1)

- Pattern recognition techniques include:
  - k-Means Clustering
  - Cosine Measure
  - Minimum Distance
  - Bayes‘ Rule
  - **Potential Function**

- A vector v has a potential function F, if

  grad(F) = v

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - **Backpropagation**
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - **Backpropagation**
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

- feed-forward multilayer network
- input for each unit = sum of outputs of previous units
- „gradient descent algorithm“ (weigths are moved along negative gradient)

How to measure „Similarity“? (1)

- Neural Network paradigms include:
  - Backpropagation
  - **Fuzzy ARTMAP**
  - Radial Basis Functions
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - Backpropagation
  - **Fuzzy ARTMAP**
  - Radial Basis Functions
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

- supervised Neural Network with very fast convergence
- comparable to Multilayer Perceptron

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - **Radial Basis Functions**
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - **Radial Basis Functions**
  - Learning Vector Quantization
  - (Hybrid) Sum-of-Products

- real-valued functions, whose values depend only on the distance from center
- popular: Gaussians

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \phi(||\mathbf{x} - \mathbf{c}_i||),$$

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - **Learning Vector Quant.**
  - (Hybrid) Sum-of-Products

How to measure „Similarity“? (I)

- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - **Learning Vector Quant.**
  - (Hybrid) Sum-of-Products

- supervised competetive network
- goal: find some kind of structure in data by determining in which way it is clustered

How to measure „Similarity“? (1)

- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - Learning Vector Quantization
  - **(Hybrid) Sum-of-Products**

How to measure „Similarity“? (I)

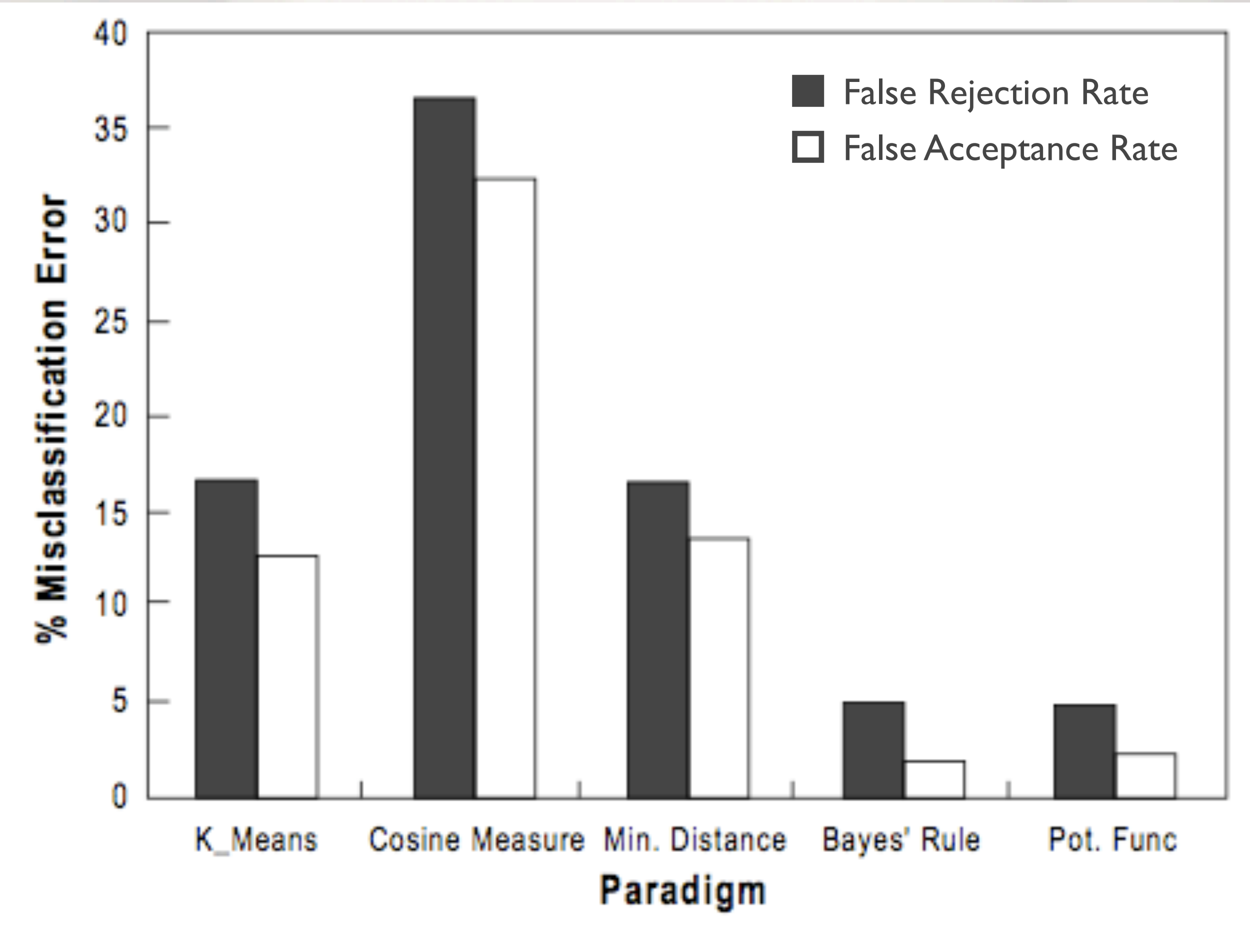
- Neural Network paradigms include:
  - Backpropagation
  - Fuzzy ARTMAP
  - Radial Basis Functions
  - Learning Vector Quantization
  - **(Hybrid) Sum-of-Products**

- backpropagation network with modified layer connections
- input for each unit = product of outputs of previous unit with weighting factor

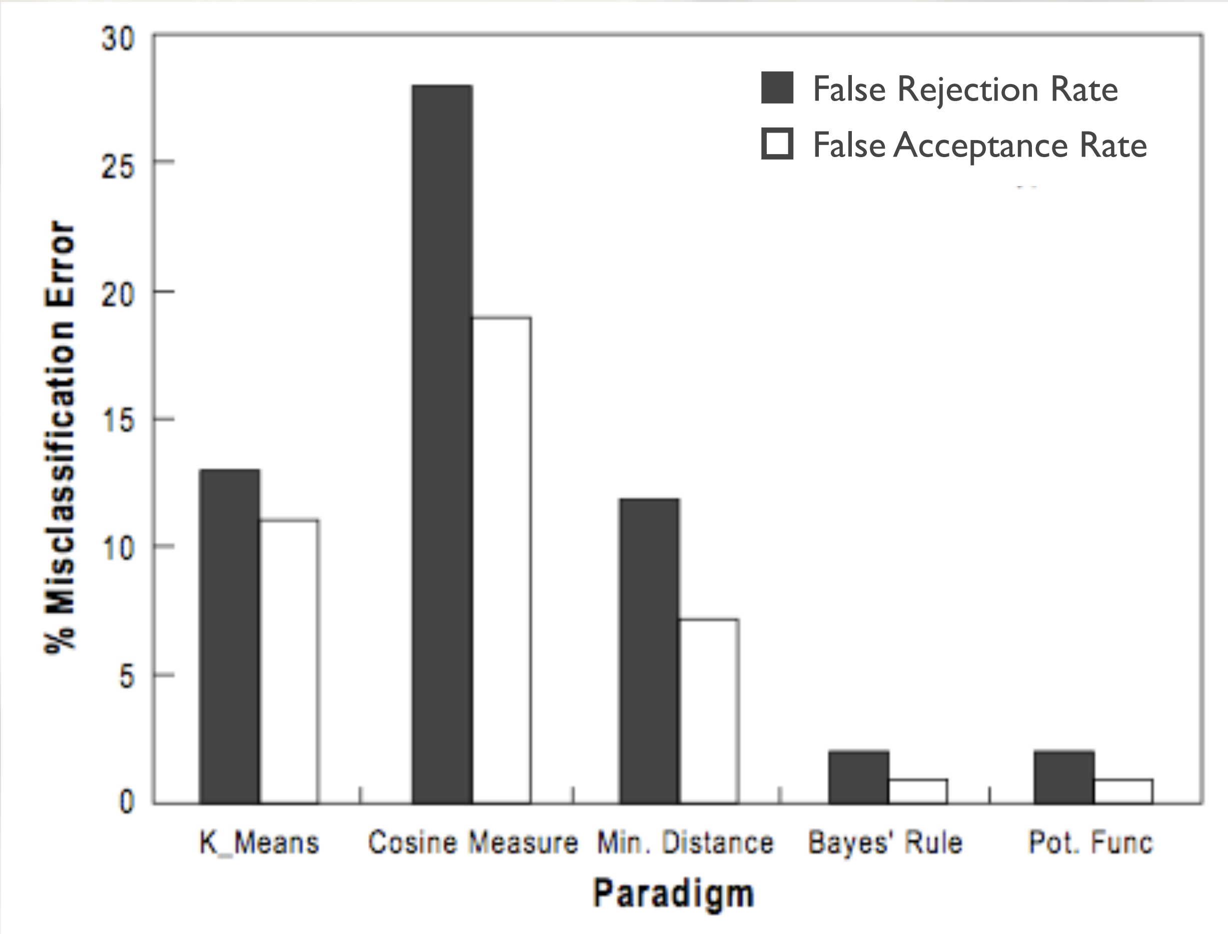
Performance

- Pattern Recognition Techniques

from Obaidat/Sadoun:
„Keystroke Dynamics based Authentication“



Classification based on **Interkey Times**

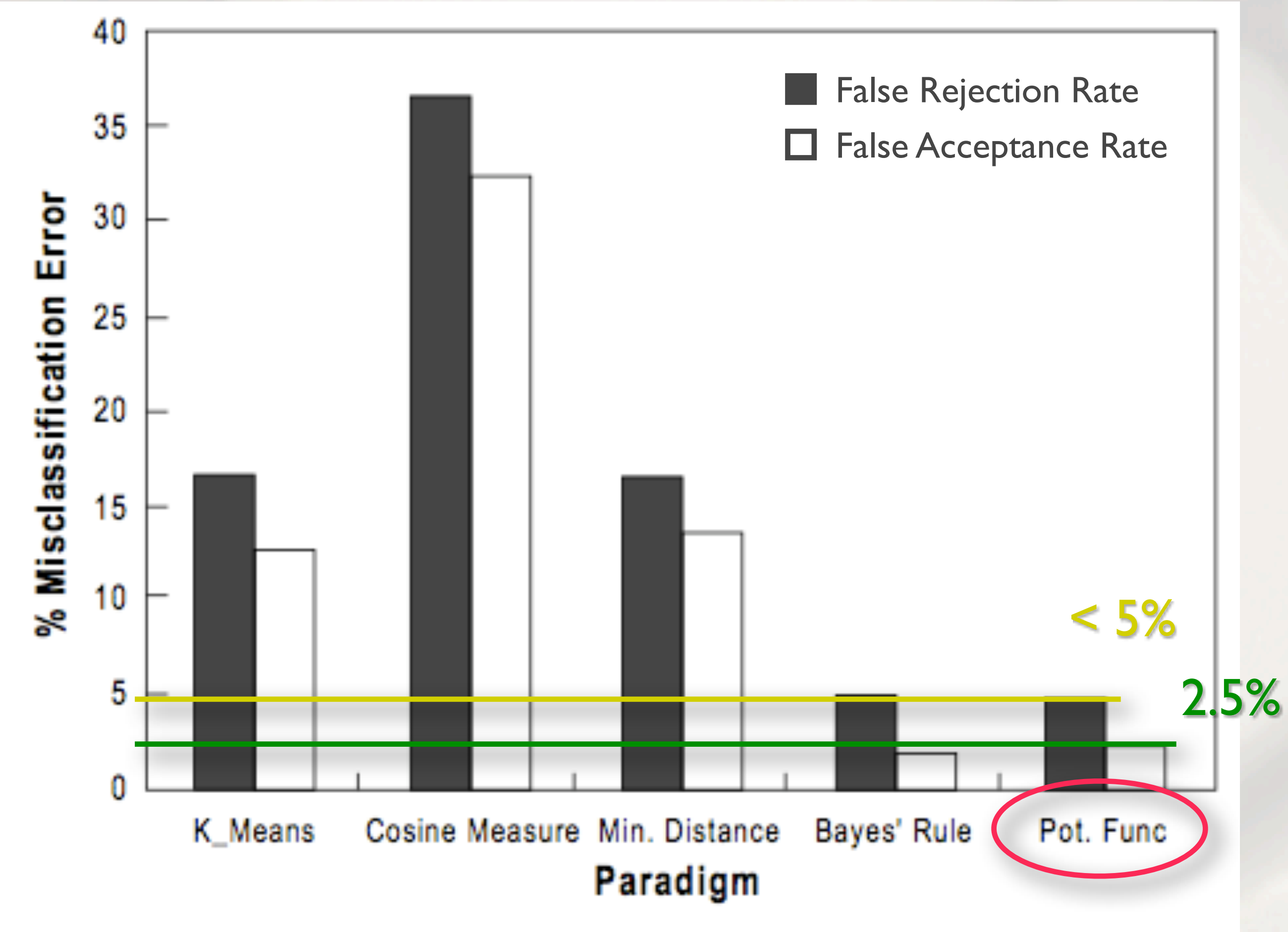


Classification based on **Hold Times**

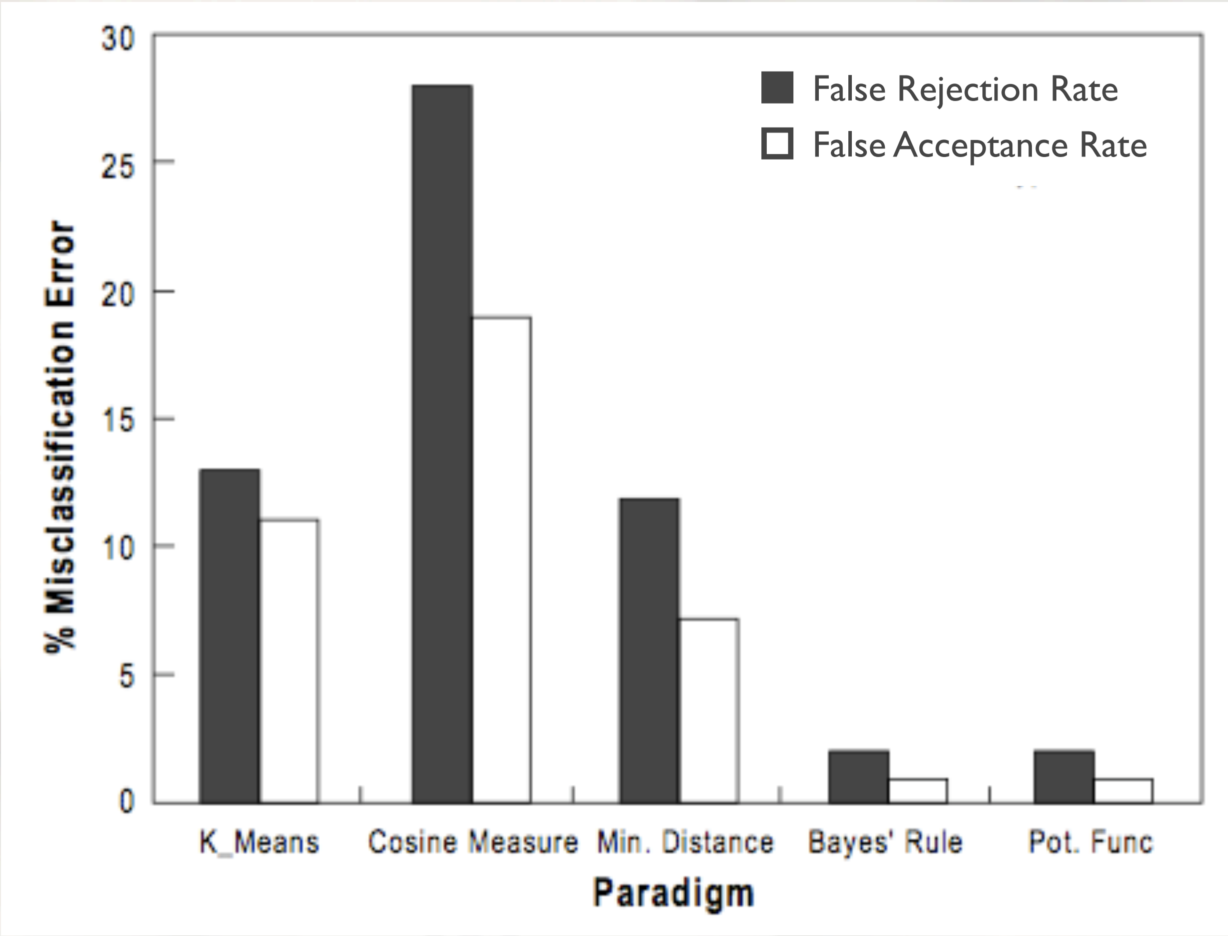
Performance

- Pattern Recognition Techniques

from Obaidat/Sadoun: **„Keystroke Dynamics based Authentication"**



Classification based on **Interkey Times**

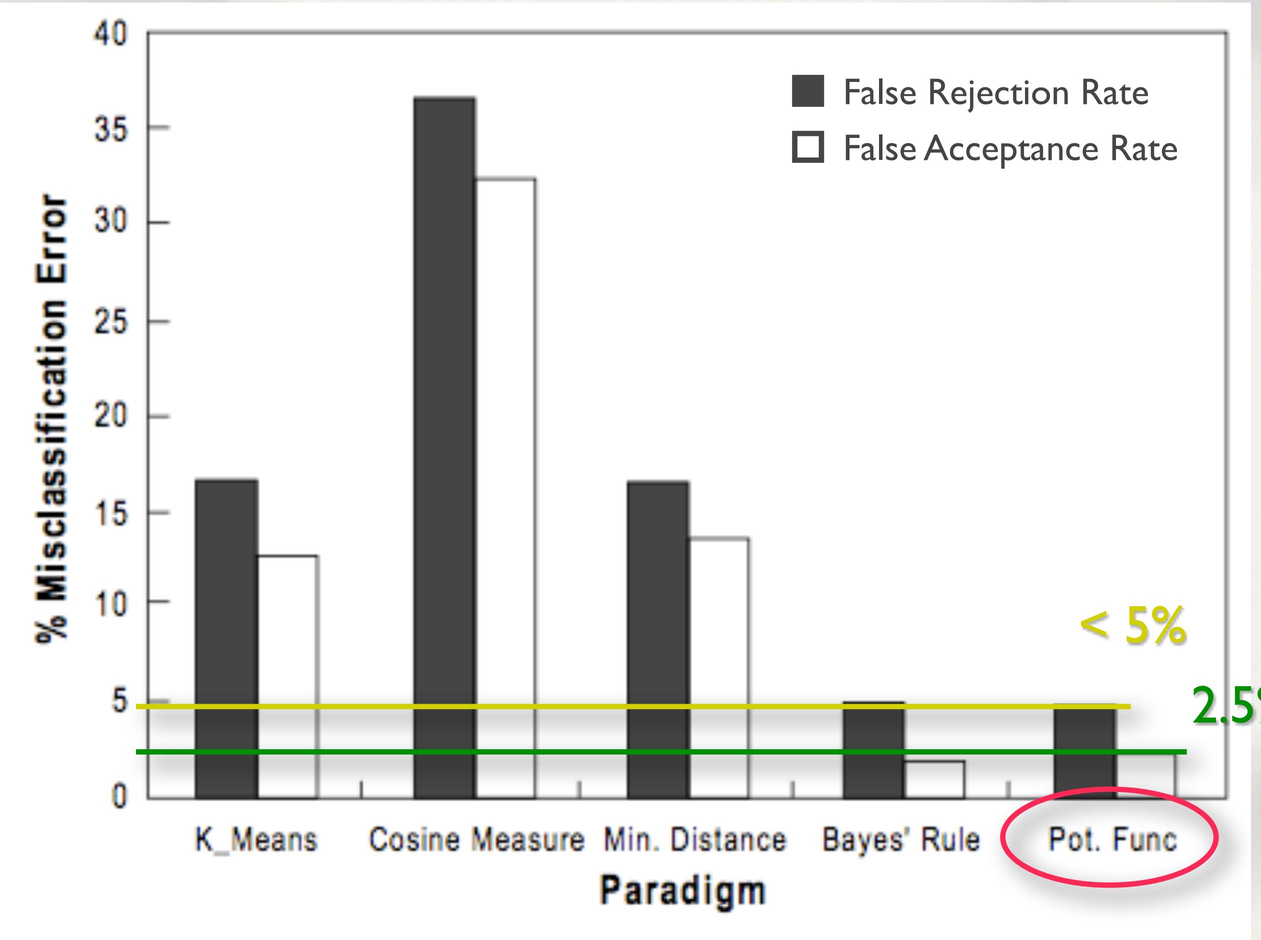


Classification based on **Hold Times**

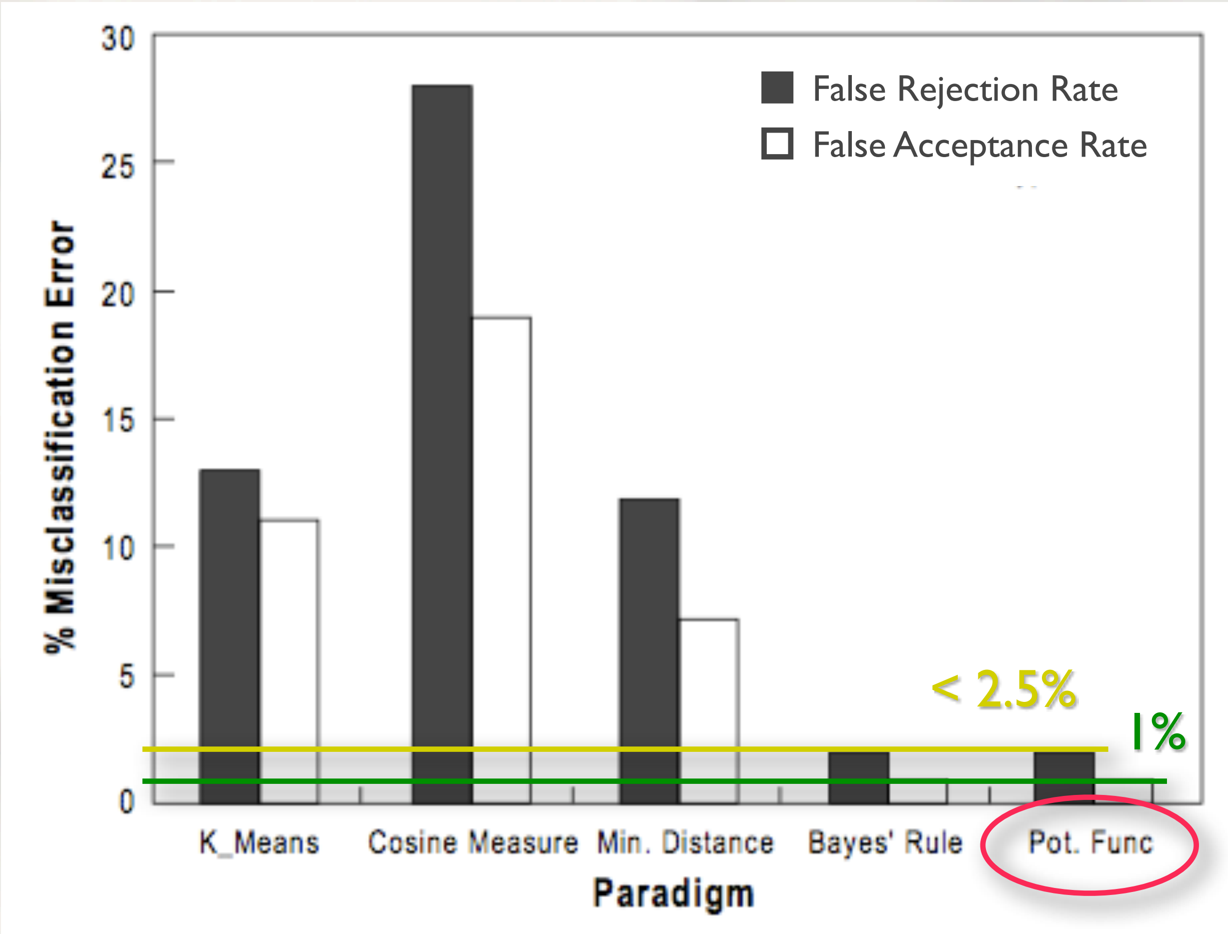
Performance

- Pattern Recognition Techniques

from Obaidat/Sadoun:
„Keystroke Dynamics based Authentication"



Classification based on **Interkey Times**

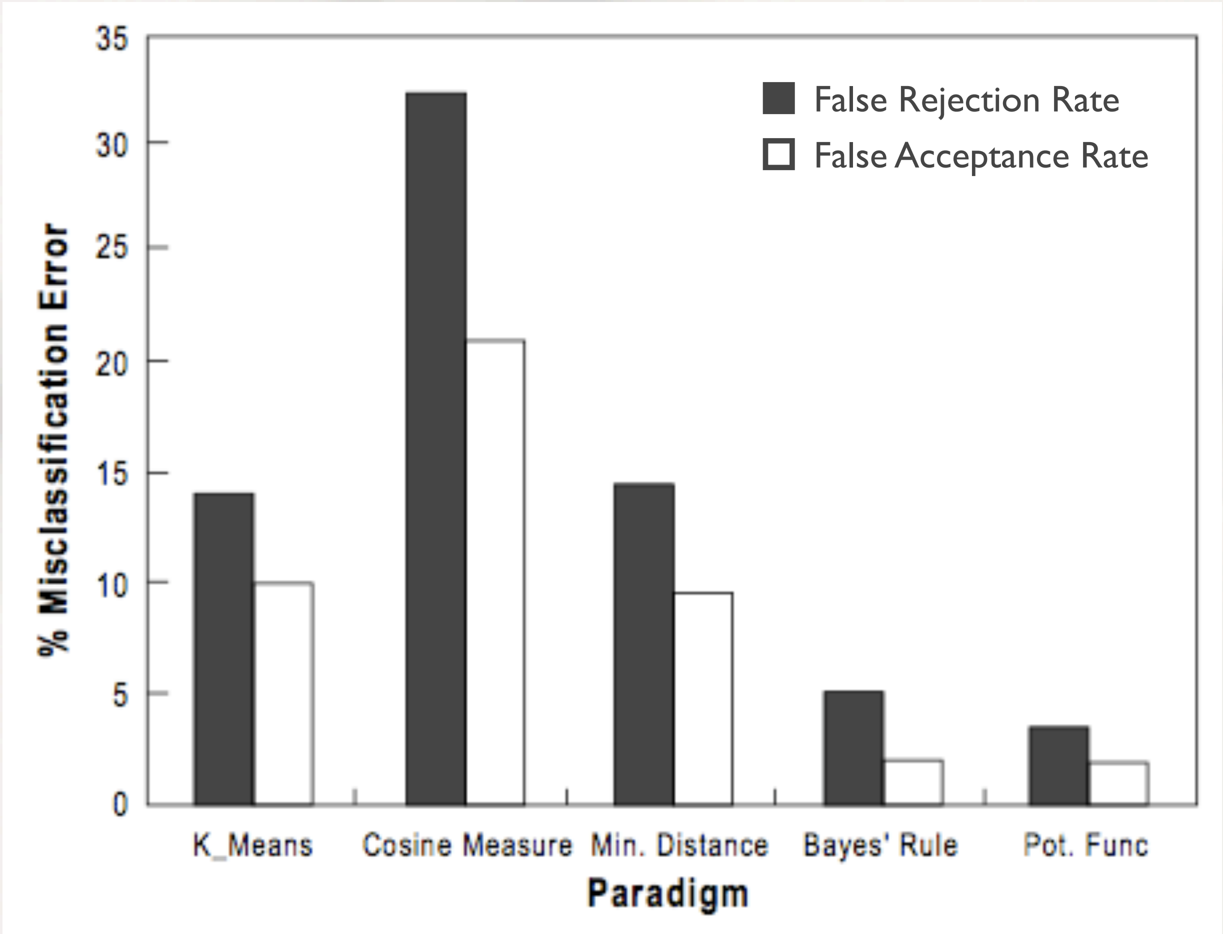


Classification based on **Hold Times**

Performance

- Pattern Recognition Techniques

from Obaidat/Sadoun: **„Keystroke Dynamics based Authentication“**

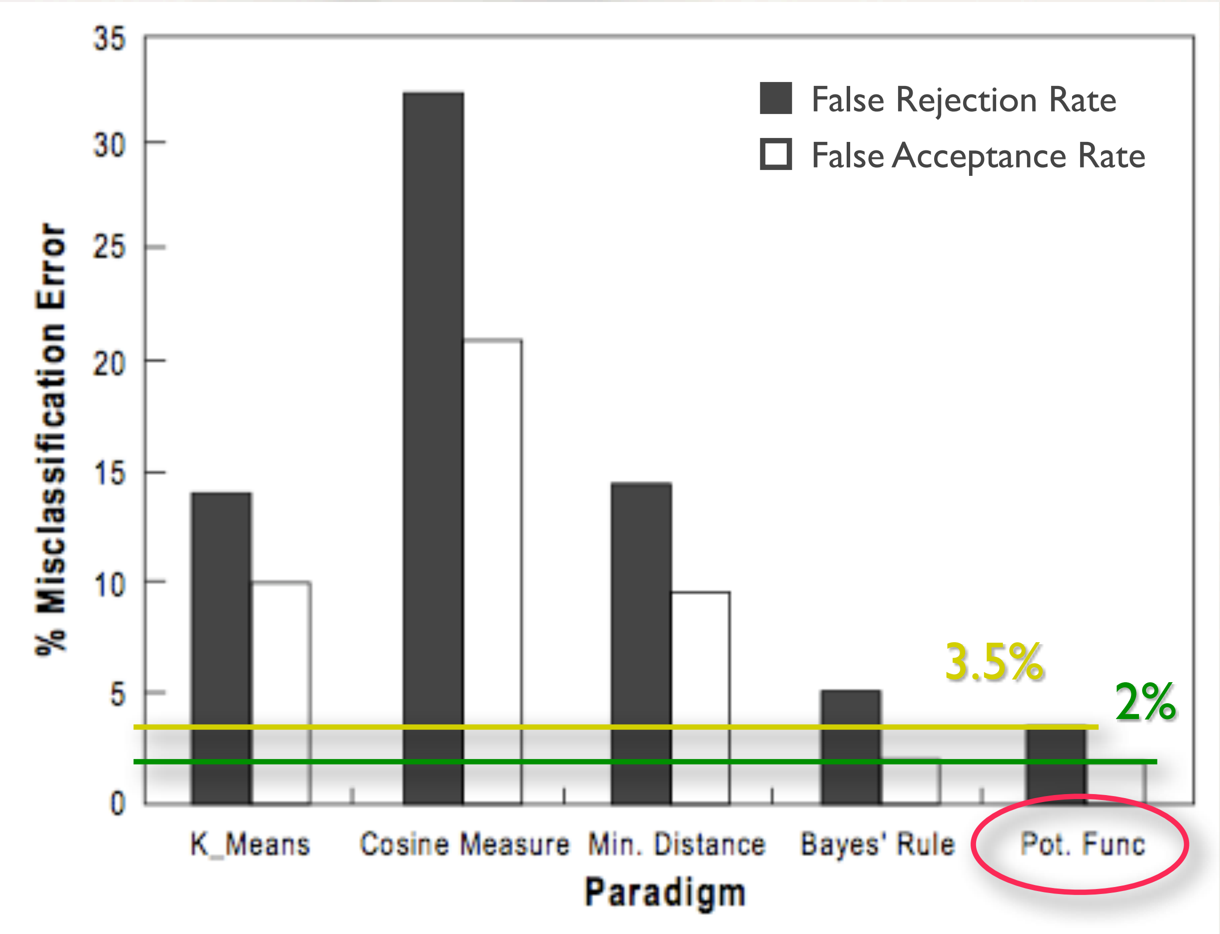


Classification based on **Hold and Interkey Times**

Performance

- Pattern Recognition Techniques

from Obaidat/Sadoun:
„Keystroke Dynamics based Authentication“

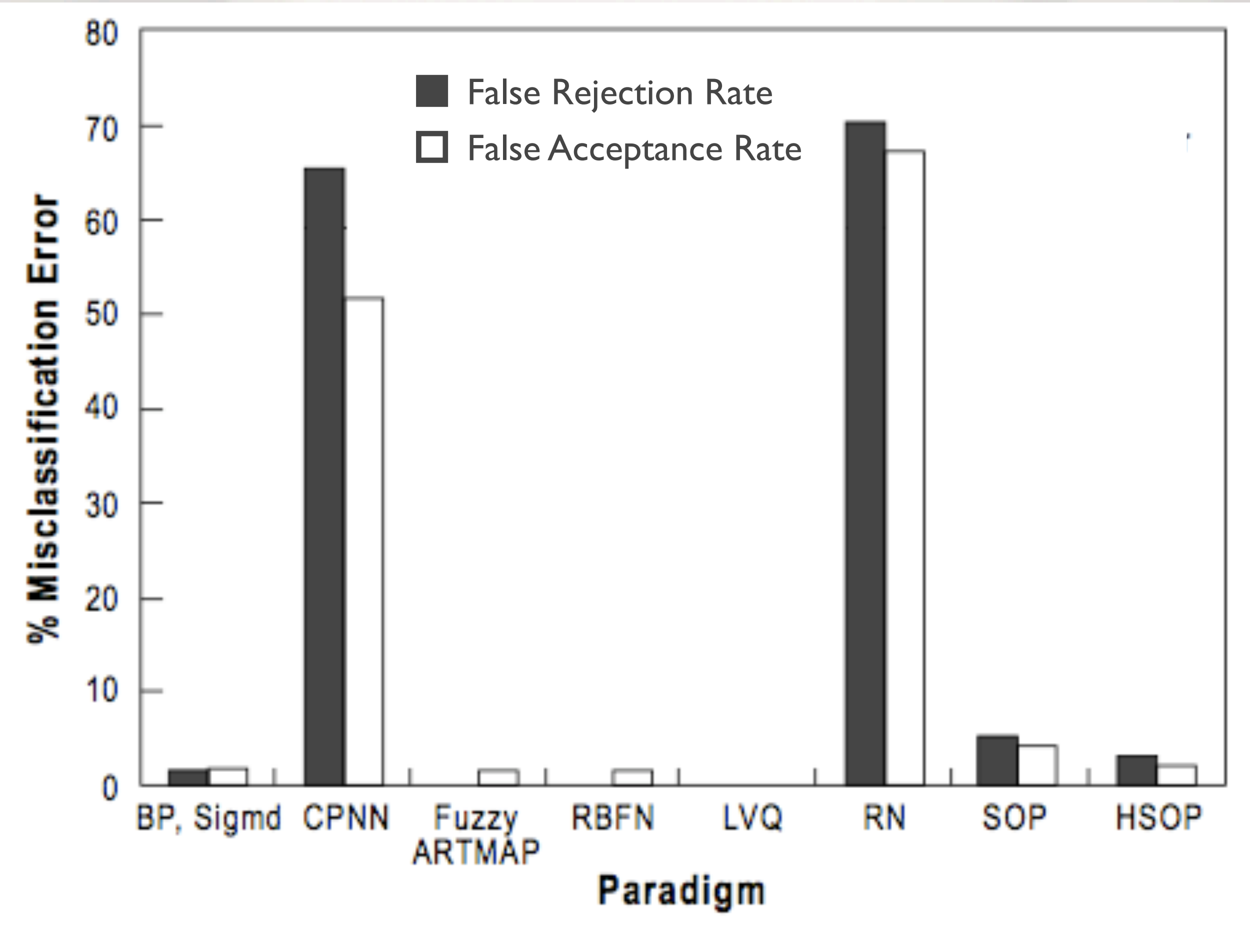


Classification based on **Hold and Interkey Times**

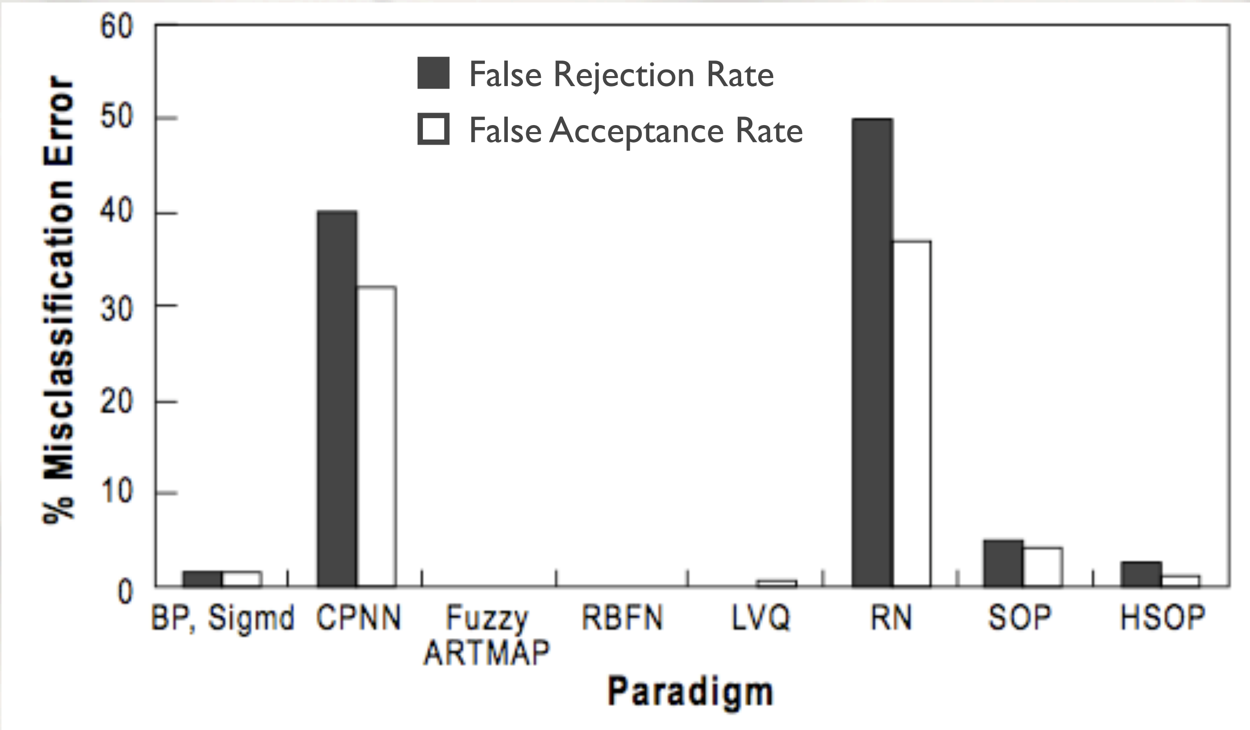
Performance

- Neural Networks

from Obaidat/Sadoun: **„Keystroke Dynamics based Authentication"**



Classification based on **Interkey Times**

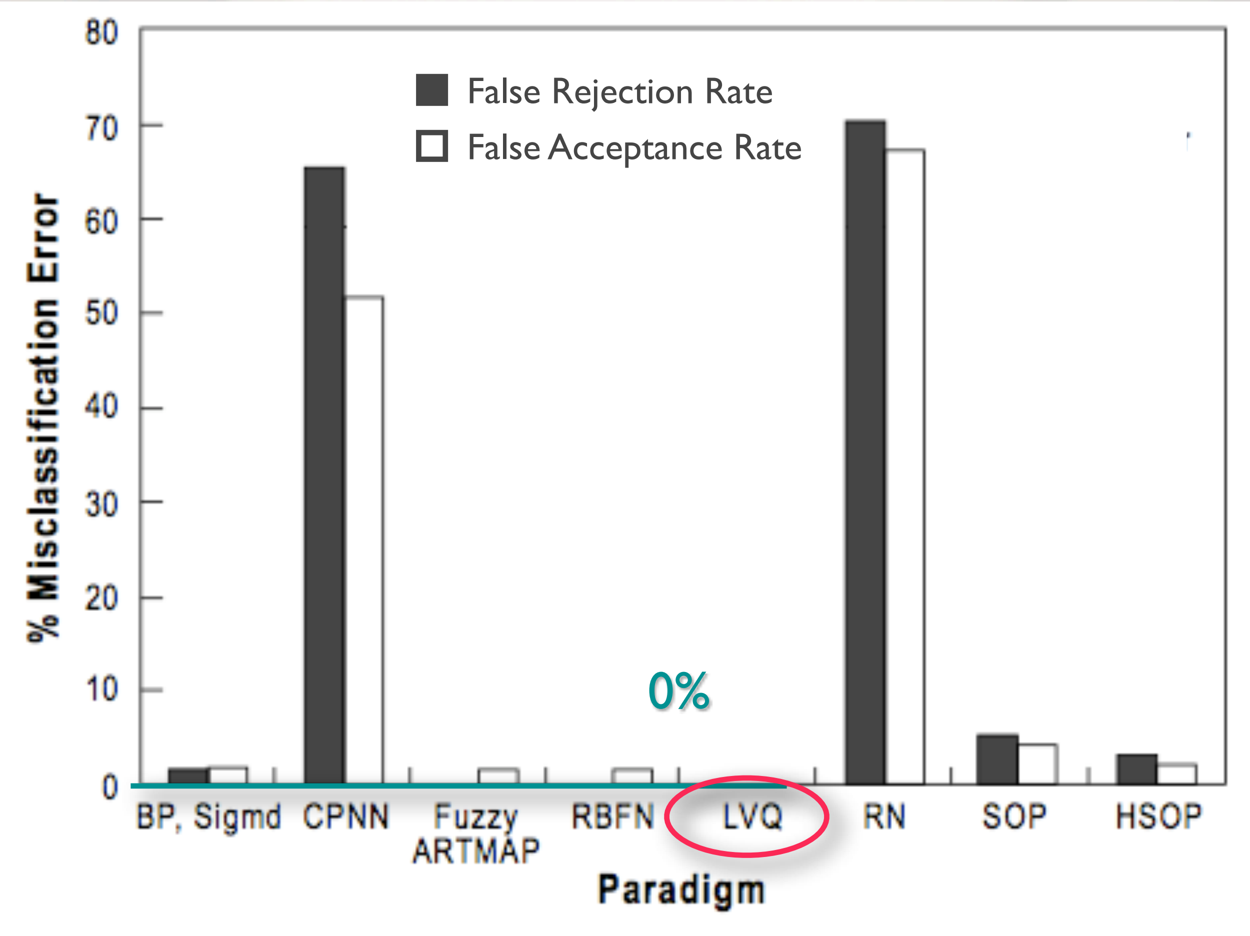


Classification based on **Hold Times**

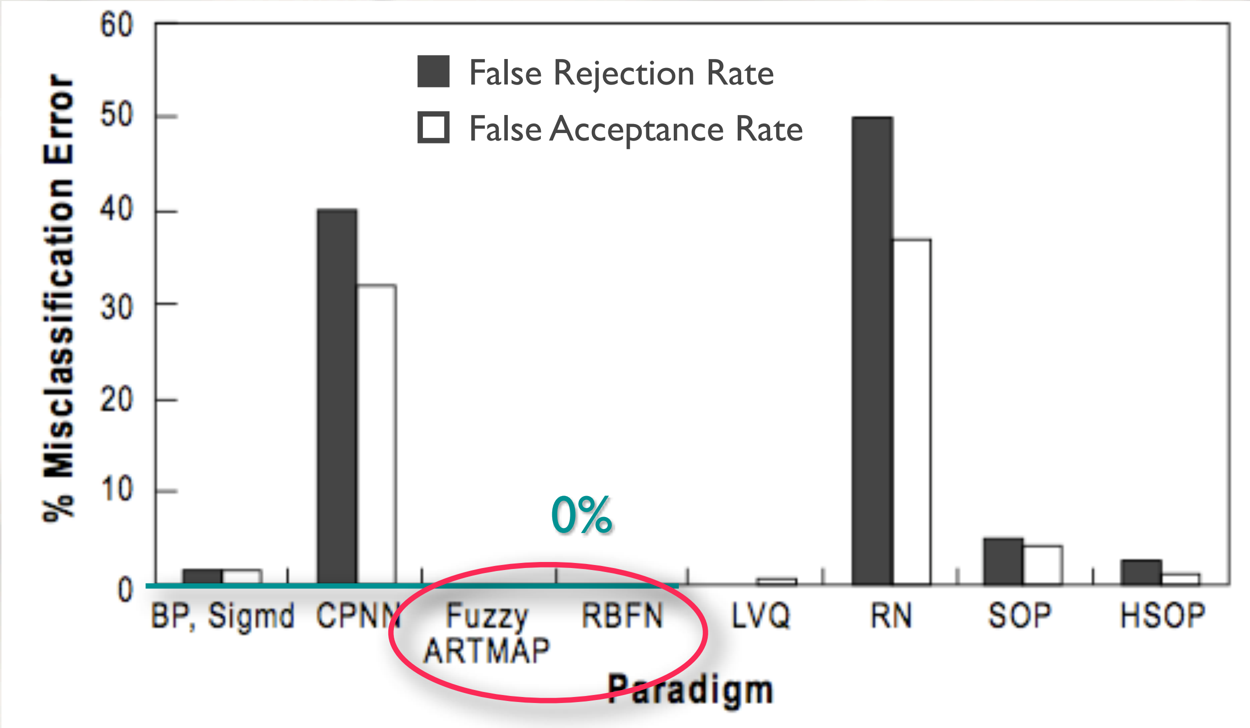
Performance

- Neural Networks

from Obaidat/Sadoun: **„Keystroke Dynamics based Authentication“**



Classification based on **Interkey Times**

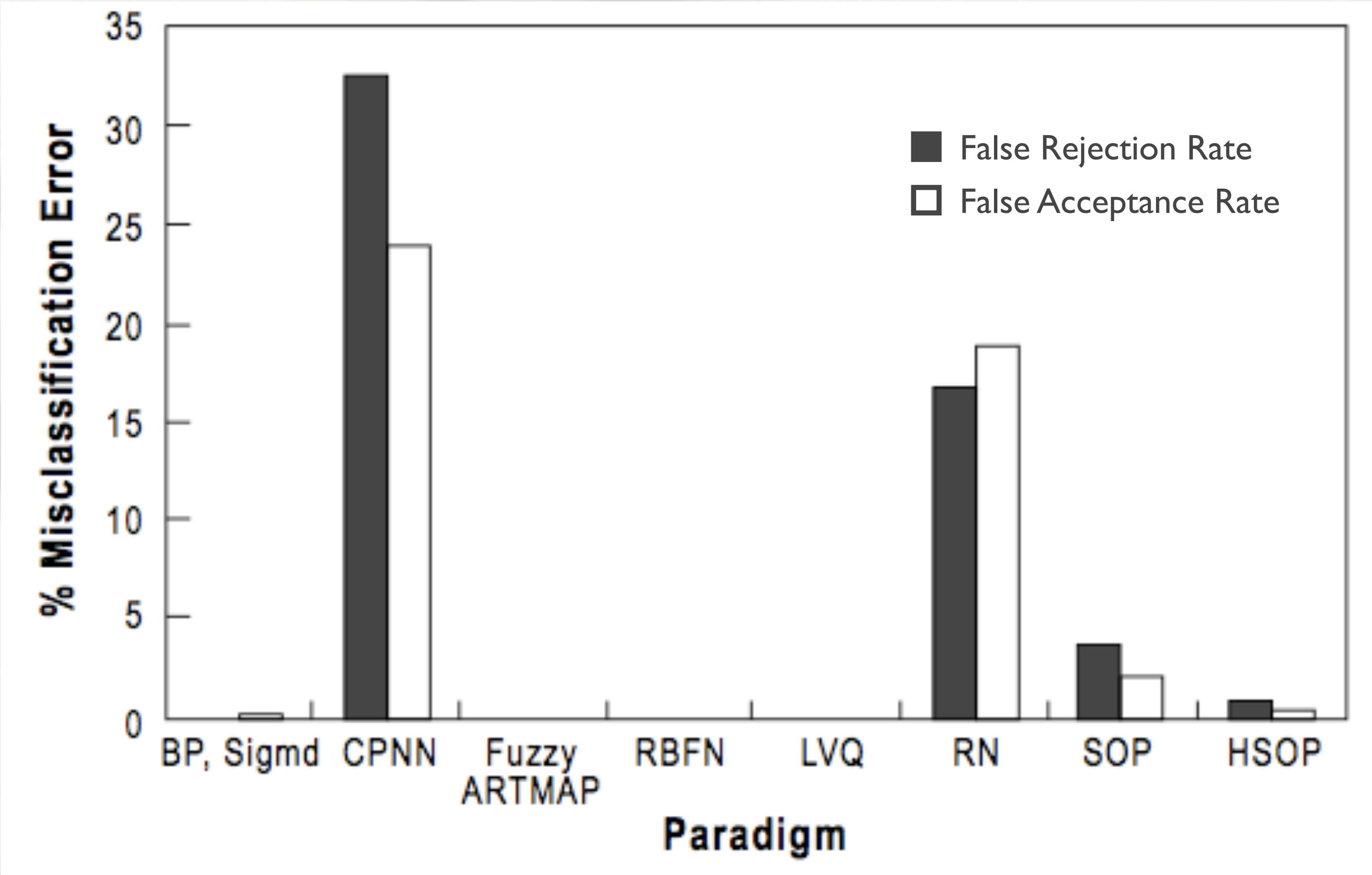


Classification based on **Hold Times**

Performance

- Neural Networks

from Obaidat/Sadoun:
„Keystroke Dynamics based Authentication“

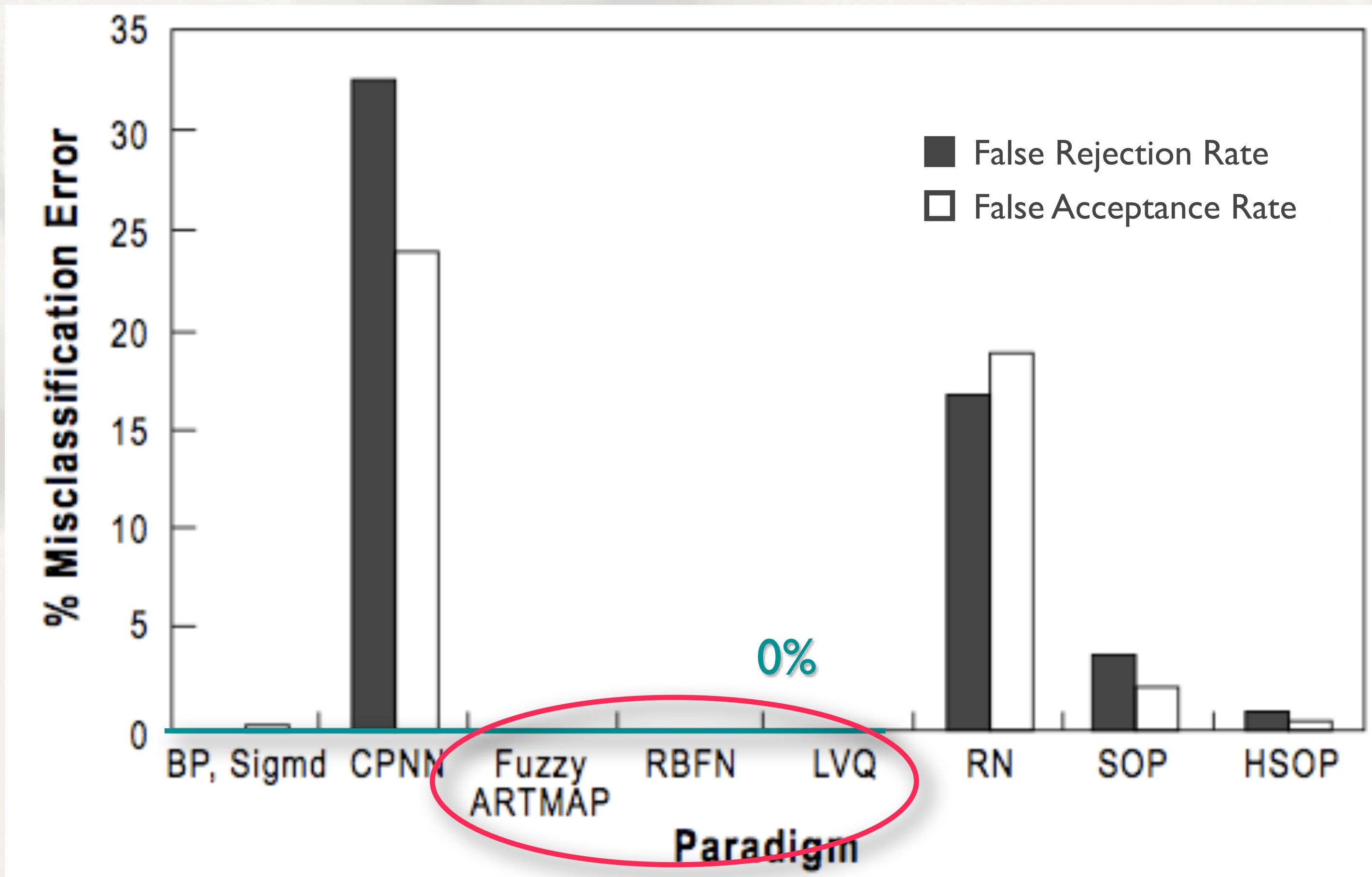


Classification based on **Hold and Interkey Times**

Performance

- Neural Networks

from Obaidat/Sadoun:
„Keystroke Dynamics based Authentication“



Classification based on **Hold and Interkey Times**

How to measure „Similarity“? (2)

- Alternative Approach (*Bergadano* et al.):
  - Measure duration (latency) of trigraphs and sort them in ascending order
  - Define distance between two samples as the „degree of disorder“ between sorted trigraphs
  - Normalize distance by maximum degree of disorder

How to measure „Similarity“? (2)

Example: User is asked to type ***austria***

Trigraph	aus	ust	str	tri	ria
Duration	277 ms	231 ms	281 ms	248 ms	295 ms

How to measure „Similarity“? (2)

<u>Example:</u> User is asked to type ***austria***

Trigraph	aus	ust	str	tri	ria
Duration	277 ms	231 ms	281 ms	248 ms	295 ms

sorted version

ust	tri	aus	str	ria
231	248	277	281	295

How to measure „Similarity“? (2)

Example: User is asked to type ***austria***

Trigraph	aus	ust	str	tri	ria
Duration	277 ms	231 ms	281 ms	248 ms	295 ms

sorted version

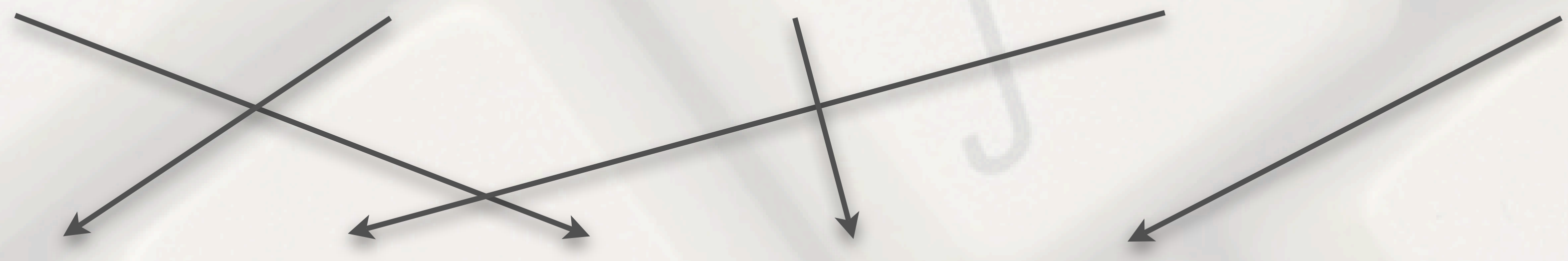
ust	tri	aus	str	ria
231	248	277	281	295

d = 1 d = 2 d = 2 d = 1 d = 0

How to measure „Similarity“? (2)

Example: User is asked to type ***austria***

Trigraph	aus	ust	str	tri	ria
Duration	277 ms	231 ms	281 ms	248 ms	295 ms



sorted version

ust	tri	aus	str	ria
231	248	277	281	295
d = 1	d = 2	d = 2	d = 1	d = 0

max. degree of disorder: $\frac{|array|^2 - 1}{2}$ → $d_{norm} = \frac{1+2+2+1+0}{12} = 0.5$

How to measure „Similarity“? (2)

Sample 1 (sorted)

ust	tri	aus	str	ria
231	248	277	281	295

How to measure „Similarity“? (2)

Sample 1 (sorted)

ust	tri	aus	str	ria
231	248	277	281	295

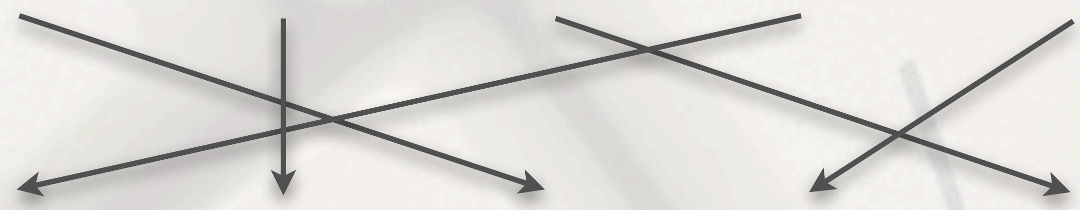
Sample 2 (sorted)

str	tri	ust	ria	aus
222	236	254	269	280
d = 3	d = 0	d = 2	d = 1	d = 2

How to measure „Similarity“? (2)

Sample 1 (sorted)

ust	tri	aus	str	ria
231	248	277	281	295



Sample 2 (sorted)

str	tri	ust	ria	aus
222	236	254	269	280
d = 3	d = 0	d = 2	d = 1	d = 2

$$d(S1, S2) = \frac{3+0+2+1+2}{12} = 0.6666$$

How to measure „Similarity“? (2)

- User Classification
  - Compute „mean distance“ of incoming sample to each user‘s model (all available samples of that user)

$$md(A, X) = (d(A_1, X) + d(A_2, X) + d(A_3, X) + d(A_4, X)) / 4$$

$$md(B, X) = (d(B_1, X) + d(B_2, X) + d(B_3, X)) / 3$$

$$md(C, X) = (d(C_1, X) + d(C_2, X) + d(C_3, X) + d(C_4, X) + d(C_5, X)) / 5$$

X .... current incoming sample

A, B, C ... user models consisting of Samples $A_{1...n}$, $B_{1...n}$, $C_{1...n}$

User Authentication

- Access Control System
  - Samples may be provided by illegal users with unknown typing patterns
  - ➡ <u>even if FAR = 0%, best possible IPR = (100/N)%</u>

- Condition:
  - Input sample must not only be closer to a certain model than to any other model; it must be <u>sufficiently</u> close to this model

User Authentication

- Access Control System
  - Samples may be provided by illegal users with unknown typing patterns

➡ even if FAR = 0%, best possible IPR = (100/N)%

„Impostor Pass Rate“ (IPR)

„False Alarm Rate“ (FAR)

- Condition:
  - Input sample must not only be closer to a certain model than to any other model; it must be sufficiently close to this model

User Authentication

- Use of Thresholds
  - Compute mean value of „inner-model“ distances:

$$m(A) = (d(A_1,A_2) + d(A_1,A_3) + d(A_1,A_4) + d(A_2,A_3) + d(A_2,A_4) + d(A_3,A_4)) / 6$$

  - Classify sample X as belonging to A, if (and only if):

$$md(A, X) < m(A) + | k * (md(B, X) - m(A)) |$$

md(B, X) ... mean value of „second closest“ model

A, B, C ... user models consisting of Samples $A_{1...n}$, $B_{1...n}$, $C_{1...n}$

User Authentication

- Values for k:
  - k = 1: plain classification scenario
  - k = 0.5: md(A, X) closer to m(A) than to any other md(B,X)
  - k = 0.33: md(A, X) twice as close to m(A) than to md(B,X)

$$md(A, X) < m(A) + | k * (md(B, X) - m(A)) |$$

md(B, X) ... mean value of „second closest" model
A, B, C ... user models consisting of Samples $A_{1...n}$, $B_{1...n}$, $C_{1...n}$

User Authentication

- Values for k:
  - k = 1: plain classification scenario
  - k = 0.5: md(A, X) closer to m(A) than to any other md(B,X)
  - k = 0.33: md(A, X) twice as close to m(A) than to md(B,X)

Value of k	$k = 1$	$k = 0.66$	$k = 0.5$	$k = 0.33$	$k = 0.3$
N. of successful attacks out of 71500 attempts	1650	98	30	2	0
N. of failed legal connections out of 220 attempts	0	0	4	13	16
Impostor Pass Rate	2.3077%	0.1371%	0.042%	0.0028%	0%
False Alarm Rate	0%	0%	1.8182%	5.9091%	7.2727%

from Bergadano et al.:
„User Authentication through Keystroke Dynamics"

md(B, X) ... mean value of „second closest" model

A, B, C ... user models consisting of Samples $A_{1...n}$, $B_{1...n}$, $C_{1...n}$

User Authentication

- Values for k:
  - k = 1: plain classification scenario
  - k = 0.5: md(A, X) closer to m(A) than to any other md(B,X)
  - k = 0.33: md(A, X) twice as close to m(A) than to md(B,X)

Value of k	$k=1$	$k=0.66$	$k=0.5$	$k=0.33$	$k=0.3$
N. of successful attacks out of 71500 attempts	1650	98	30	2	0
N. of failed legal connections out of 220 attempts	0	0	4	13	16
Impostor Pass Rate	2.3077%	0.1371%	0.042%	0.0028%	0%
False Alarm Rate	0%	0%	1.8182%	5.9091%	7.2727%

from Bergadano et al.:
„User Authentication through Keystroke Dynamics"

md(B, X) ... mean value of „second closest" model

A, B, C ... user models consisting of Samples $A_{1...n}$, $B_{1...n}$, $C_{1...n}$

User Authentication

- Additional filtering
  - Thresholds improve classification results but may have counterintuitive behavior:

$$md(A, X) = 0.307025 \qquad d(A_1, A_2) = 0.212378$$
$$md(B, X) = 0.420123 \qquad d(A_1, A_3) = 0.204381$$
$$md(C, X) = 0.423223 \qquad d(A_2, A_3) = 0.226024$$
$$m(A) = 0.214261$$

$$md(A, X) < m(A) + | k * (md(B, X) - m(A)) |$$

User Authentication

- Additional filtering
  - Thresholds improve classification results but may have counterintuitive behavior:

$$md(A, X) = 0.307025 \qquad d(A_1, A_2) = 0.212378$$
$$md(B, X) = 0.420123 \qquad d(A_1, A_3) = 0.204381$$
$$md(C, X) = 0.423223 \qquad d(A_2, A_3) = 0.226024$$
$$m(A) = 0.214261$$

$$0.307025 < 0.214261 + | 0.5 * (0.420123 - 0.214261) |$$ ✓

User Authentication

- Additional filtering
  - Thresholds improve classification results but may have counterintuitive behavior:

$md(A, X) = 0.307025$
$md(B, X) = 0.420123$
$md(C, X) = 0.423223$

$d(A_1, A_2) = 0.212378$
$d(A_1, A_3) = 0.204381$
$d(A_2, A_3) = 0.226024$

$m(A) = 0.214261$

?

$0.307025 < 0.214261 + | 0.5 * (0.420123 - 0.214261) |$ ✓

User Authentication

- Additional filtering
  - For each sample, compute mean distance w.r.t. all the other samples in the model:

$$m(A_{xyz}) = (d(A_x,A_y) + d(A_x,A_z) + d(A_y,A_z)) / 3$$

$$dA_1 = | (d(A_1,A_2) + d(A_1,A_3) + d(A_1,A_4)) / 3 - m(A_{234}) |$$
$$dA_2 = | (d(A_2,A_1) + d(A_2,A_3) + d(A_2,A_4)) / 3 - m(A_{134}) |$$
$$dA_3 = | (d(A_3,A_1) + d(A_3,A_2) + d(A_3,A_4)) / 3 - m(A_{124}) |$$
$$dA_4 = | (d(A_4,A_1) + d(A_4,A_2) + d(A_4,A_3)) / 3 - m(A_{123}) |$$

User Authentication

- Additional filtering
  - For each sample, compute mean distance w.r.t. all the other samples in the model:

$$m(A_{xyz}) = (d(A_x,A_y) + d(A_x,A_z) + d(A_y,A_z)) / 3$$

$$dA_1 = | (d(A_1,A_2) + d(A_1,A_3) + d(A_1,A_4)) / 3 - m(A_{234}) |$$
$$dA_2 = | (d(A_2,A_1) + d(A_2,A_3) + d(A_2,A_4)) / 3 - m(A_{134}) |$$
$$dA_3 = | (d(A_3,A_1) + d(A_3,A_2) + d(A_3,A_4)) / 3 - m(A_{124}) |$$
$$dA_4 = | (d(A_4,A_1) + d(A_4,A_2) + d(A_4,A_3)) / 3 - m(A_{123}) |$$

$$md(A, X) < m(A) + a * max(dA_1, dA_2, dA_3, dA_4) + b * std(dA_1, dA_2, dA_3, dA_4)$$

User Authentication

- Additional filtering
  - For each sample, compute mean distance w.r.t. all the other samples in the model:

value of k value of a value of b	$k=0.5$ $a=1$ $b=1.5$	$k=0.5$ $a=1$ $b=1.75$	$k=0.5$ $a=1.5$ $b=0$	$k=0.5$ $a=1.5$ $b=0.5$	no k $a=1.5$ $b=0.5$	$k=0.55$ $a=1.22$ $b=1.25$
Successful attacks (out of 71500)	3	5	4	7	1032	7
Failed legal connections (out of 220)	12	9	10	8	5	4
IPR FAR	0.0042% 5.4545%	0.007% 4.0909%	0.0056% 4.5454%	0.0098% 3.6364%	1.4433% 2.2727%	0.0098% 1.8182%

from Bergadano et al.:
„User Authentication through Keystroke Dynamics“

$$md(A, X) < m(A) + a * \max(dA_1, dA_2, dA_3, dA_4) + b * std(dA_1, dA_2, dA_3, dA_4)$$

User Authentication

- Additional filtering
  - For each sample, compute mean distance w.r.t. all the other samples in the model:

value of k value of a value of b	$k=0.5$ $a=1$ $b=1.5$	$k=0.5$ $a=1$ $b=1.75$	$k=0.5$ $a=1.5$ $b=0$	$k=0.5$ $a=1.5$ $b=0.5$	no k $a=1.5$ $b=0.5$	$k=0.55$ $a=1.22$ $b=1.25$
Successful attacks (out of 71500)	3	5	4	7	1032	7
Failed legal connections (out of 220)	12	9	10	8	5	4
IPR FAR	0.0042% 5.4545%	0.007% 4.0909%	0.0056% 4.5454%	0.0098% 3.6364%	1.4433% 2.2727%	0.0098% 1.8182%

from Bergadano et al.: **„User Authentication through Keystroke Dynamics“**

$$md(A, X) < m(A) + a * \max(dA_1, dA_2, dA_3, dA_4) + b * \mathrm{std}(dA_1, dA_2, dA_3, dA_4)$$

How to measure „Similarity“? (2)

- Advantages
  - Measure considers relative relative values of various typing features only
  - No need for specific tuning or training
  - Typing errors allowed (additional pre-filtering to keep only the shared trigraphs)

Bibliography

- „User Authentication through Keystroke Dynamics“
  F. Bergadano et al.
- „Keystroke Dynamics based Authentication“
  M.S. Obaidat and **B. Sadoun**
- „Artificial Rhythms and Cues for Keystroke Dynamics based Authentication“
  S. Cho and **S. Hwang**
- „Computer User Authentication using Hidden Markov Model through Keystroke Dynamics “
  S.K. Vuyyuru et al.

Thank you.