Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks

William Melicher, Blase Ur, Sean Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor

First appeared at: USENIX Security 2016

Carnegie Mellon University

Guessing Methods

Guessing Methods

- **John the Ripper + Hashcat**

Guessing Methods

- **John the Ripper + Hashcat**

Dictionary word + Rules

Guessing Methods

- **John the Ripper + Hashcat**

Dictionary word + Rules

`password` + append 2 digits

Guessing Methods

- **John the Ripper + Hashcat**

Dictionary word + Rules

`password` + append 2 digits

```
password11
password12
...
```

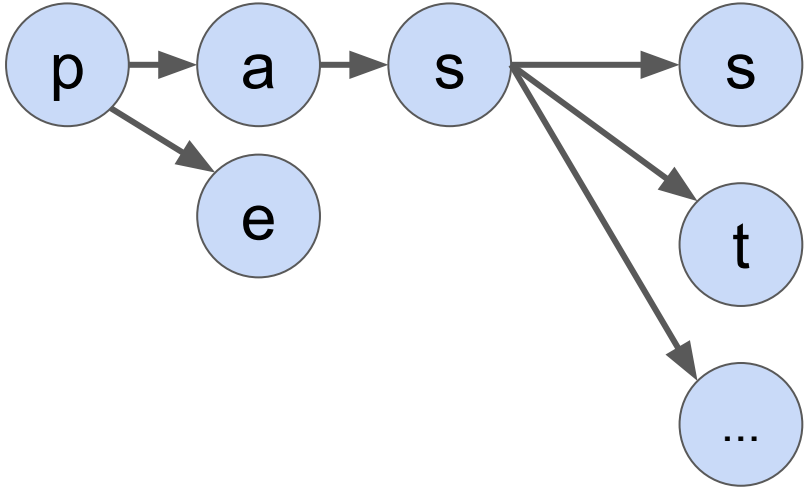
Guessing Methods

- **John the Ripper + Hashcat**
- **Markov Models**

Guessing Methods

- **John the Ripper + Hashcat**

- **Markov Models**

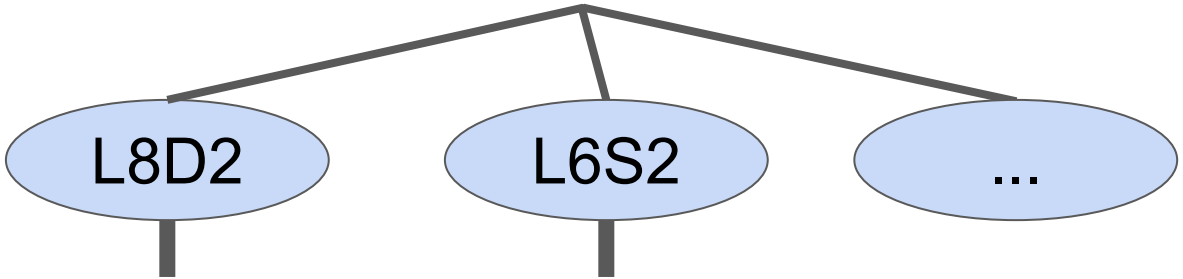


Guessing Methods

- **John the Ripper + Hashcat**
- **Markov Models**
- **PCFGs**

Guessing Methods

- **John the Ripper + Hashcat**
- **Markov Models**
- **PCFGs**

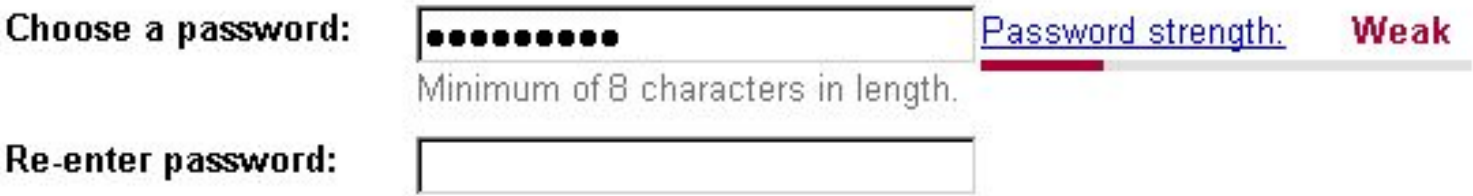



```
password12   monkey!!
password11   qwerty..
...          ...
```

Guessing Methods

- **John the Ripper + Hashcat**
- **Markov Models**
- **PCFGs**

Why Model Guessing Attacks?



Can we guess more accurately?

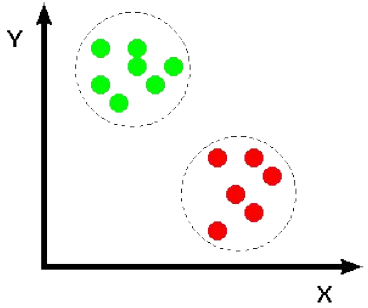
Quicker?

With fewer resources?

Our Approach: Neural Networks

Hello = Здравствуйте

Handwriting Recognition →
Handwriting recognition



Outline: Guessing with Neural Networks

- Password guesser design
- Comparison to other methods
- Real-time, in-browser feedback

Generating Passwords by Predicting

Generating Passwords by Predicting

`passw` → o or maybe `0` or 0 or ...

Generating Passwords by Predicting

passw →

Next char is:

A: 3%
B: 1%
C: 0.6%
…
O: 55%
…
Z: 0.01%
0: 20%
1: ...

Generating Passwords by Predicting

“”

P: 100%

Generating Passwords by Predicting

“”

P: 100%

Next char is:

A:	3%
B:	2%
C:	5%
…	
O:	2%
…	
Z:	0.2%
0:	1%
1:	…
END:	2%

Generating Passwords by Predicting

“”

P: 100%

Next char is:

A:	3%
B:	2%
C:	5%
…	
O:	2%
…	
Z:	0.2%
0:	1%
1:	…
END:	2%

Generating Passwords by Predicting

“”

P: 100%

Next char is:

A:	3%
B:	2%
C:	5%
…	
O:	2%
…	
Z:	0.2%
0:	1%
1:	…
END:	2%

Generating Passwords by Predicting

“C”
P: 5%

Generating Passwords by Predicting

“C”

P: 5%

Next char is:

A:	10%
B:	1%
C:	4%
…	
O:	8%
…	
Z:	0.02%
0:	3%
1:	…
END:	6%

Generating Passwords by Predicting

"C"

P: 5%

Next char is:

A:	10%
B:	1%
C:	4%
…	
O:	8%
…	
Z:	0.02%
0:	3%
1:	…
END:	6%

Generating Passwords by Predicting

"CA"
P: 0.5%

Next char is:

A:	3%
B:	10%
C:	7%
…	
O:	1%
…	
Z:	0.03%
0:	2%
1:	…
END:	12%

Generating Passwords by Predicting

"CAB"

P: 0.05%

Next char is:

A:	3%
B:	10%
C:	7%
…	
O:	1%
…	
Z:	0.03%
0:	2%
1:	…
END:	3%

Generating Passwords by Predicting

"CAB"
P: 0.05%

Next char is:

A:	4%
B:	3%
C:	1%
…	
O:	2%
…	
Z:	0.01%
0:	4%
1:	…
END:	12%

Generating Passwords by Predicting

"CAB"

P: 0.05%

Next char is:

A:	4%
B:	3%
C:	1%
…	
O:	2%
…	
Z:	0.01%
0:	4%
1:	…
END:	12%

Generating Passwords by Predicting

“CAB”
P: 0.006%

Generating Passwords

`CAB` - 0.006%
`CAC` - 0.0042%
`ADD1` - 0.002%
`CODE` - 0.0013%
...

Generating Passwords

CAR - 0.006%

CAC - 0.0042%

ADD1 - 0.002%

CODE - 0.0013%

...

MUST BE LONGER THAN 3 CHARACTERS

Password Policies: 1class8

1 character class and 8 characters minimum

`password123`

`12345678`

`monkey99`

Password Policies: 3class12

3 character class and 12 characters minimum

llamalove123

Mypassword#3

N@rut0_r0ck5

Outline: Guessing with Neural Networks

- Password guesser design
- **Comparison to other methods**
- Real-time, in-browser feedback

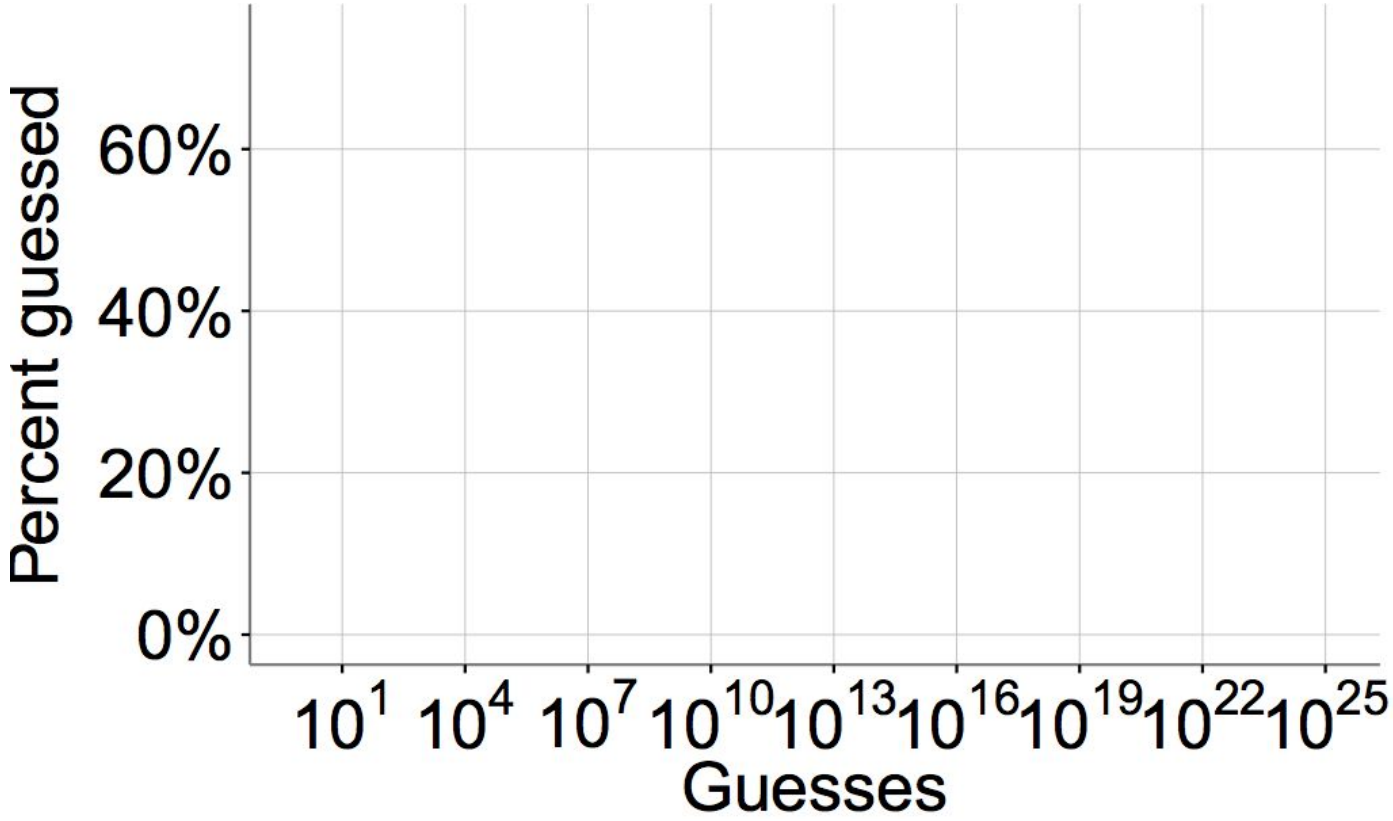
We Had to Try Many Parameters

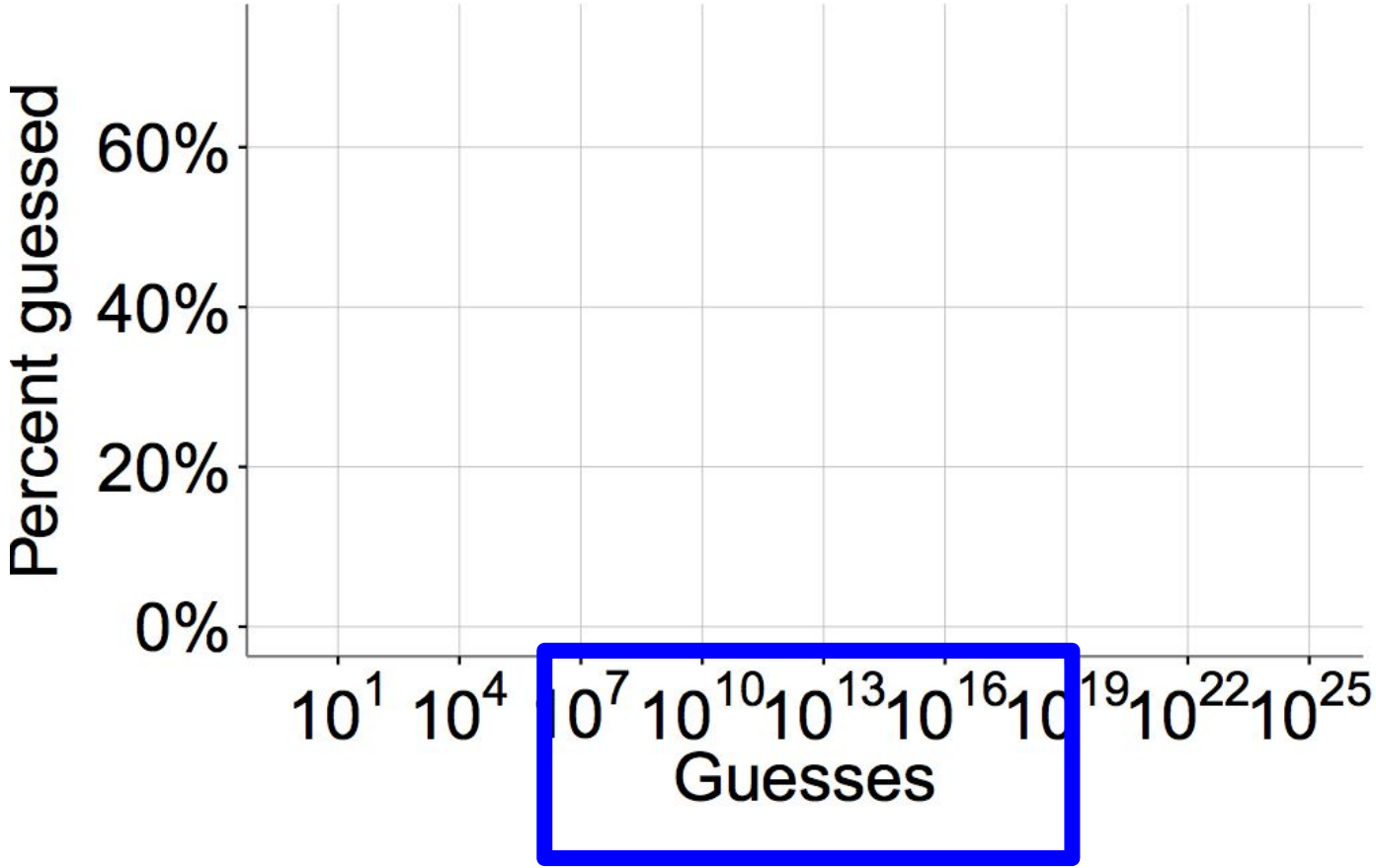
- Model size: 60MB, 3MB
- Transference learning
- Training data
- Model architecture
- Alphabet size
- Password context

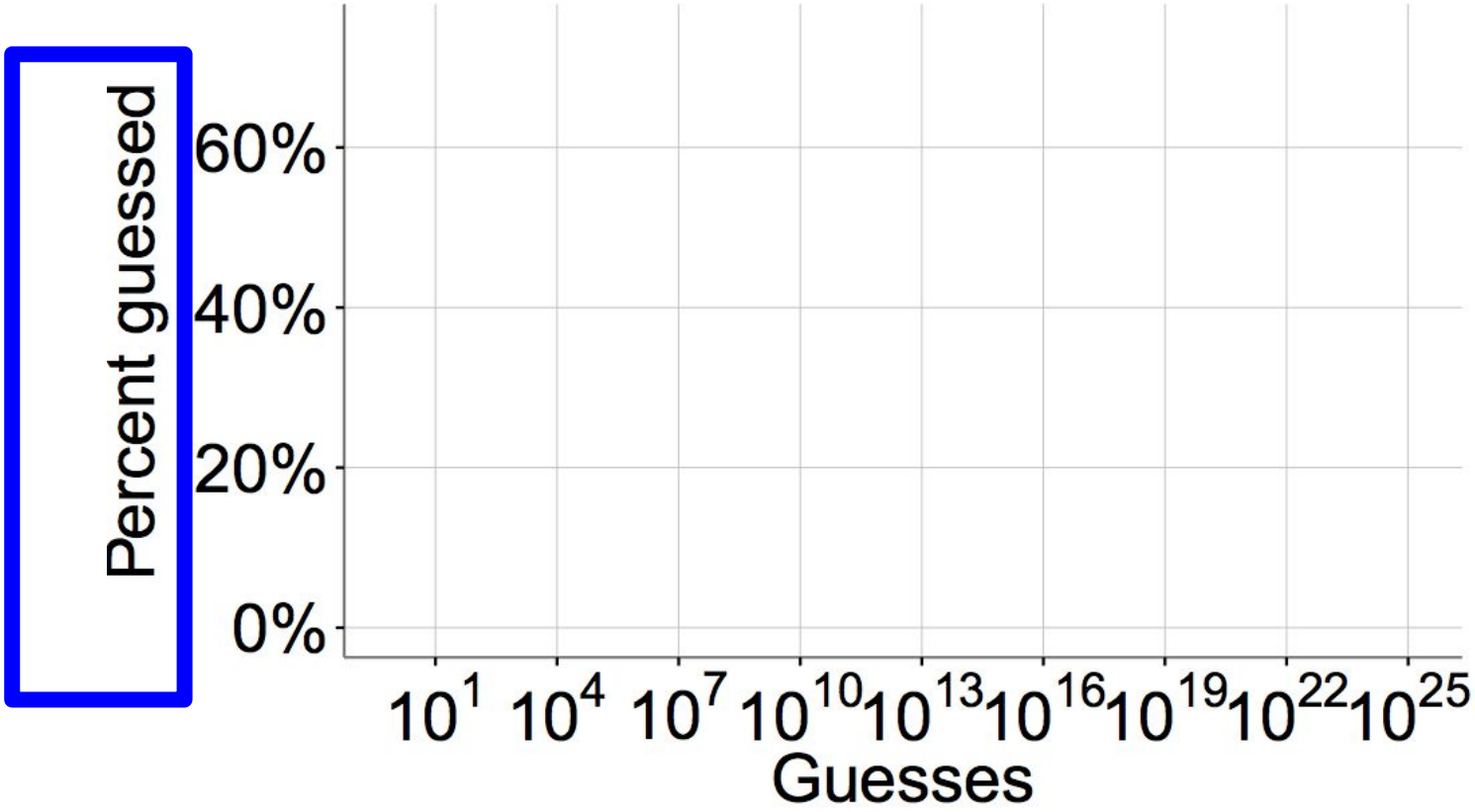
Testing Methodology

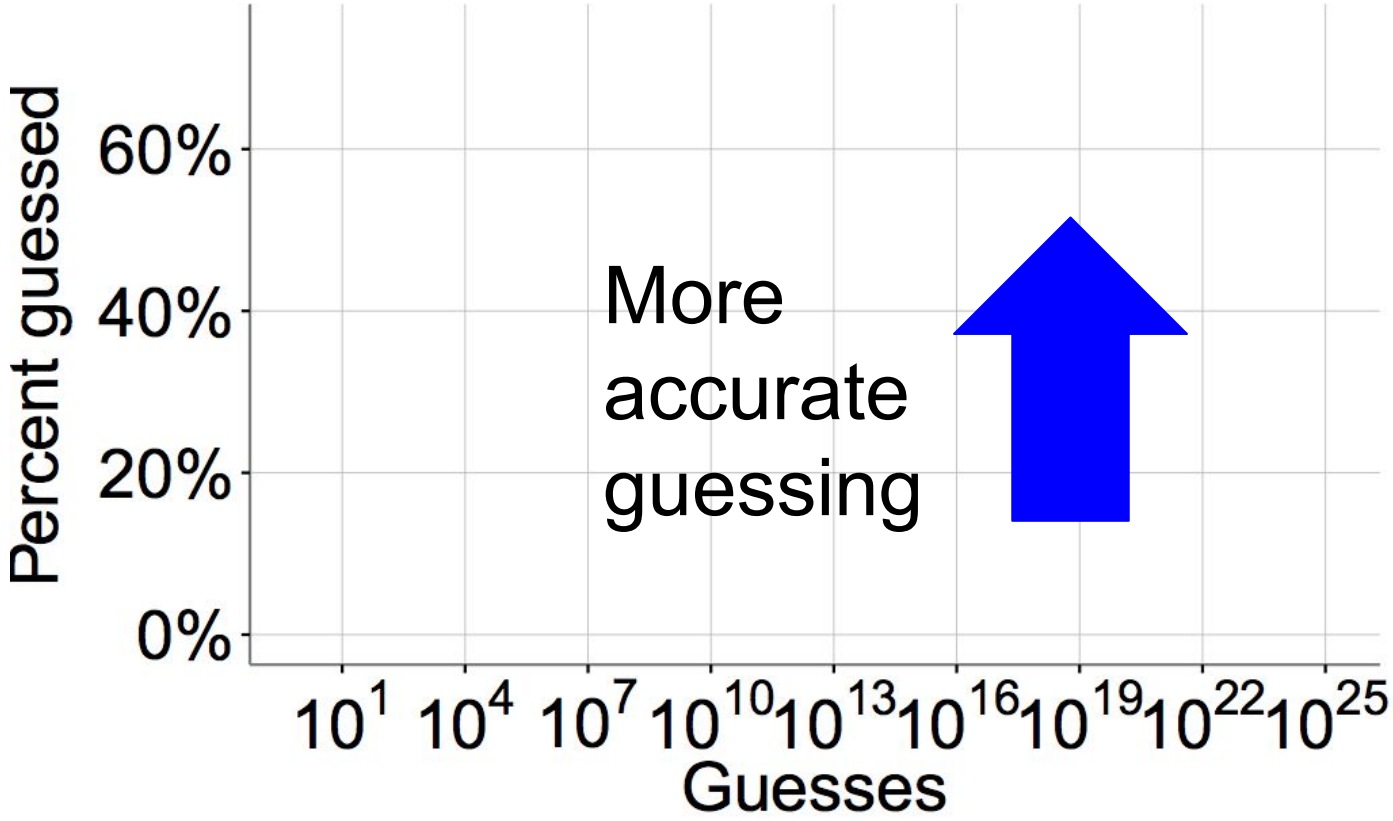
- Approach: measure # guessed passwords
- Training data: leaked password sets
- Testing data
  - MTurk study passwords: 1class8, 4class8, 1class16, 3class12
  - Real passwords: 000webhost password leak
- Estimate guess numbers with Monte-Carlo technique (Dell'Amico and Filippone, CCS '15)

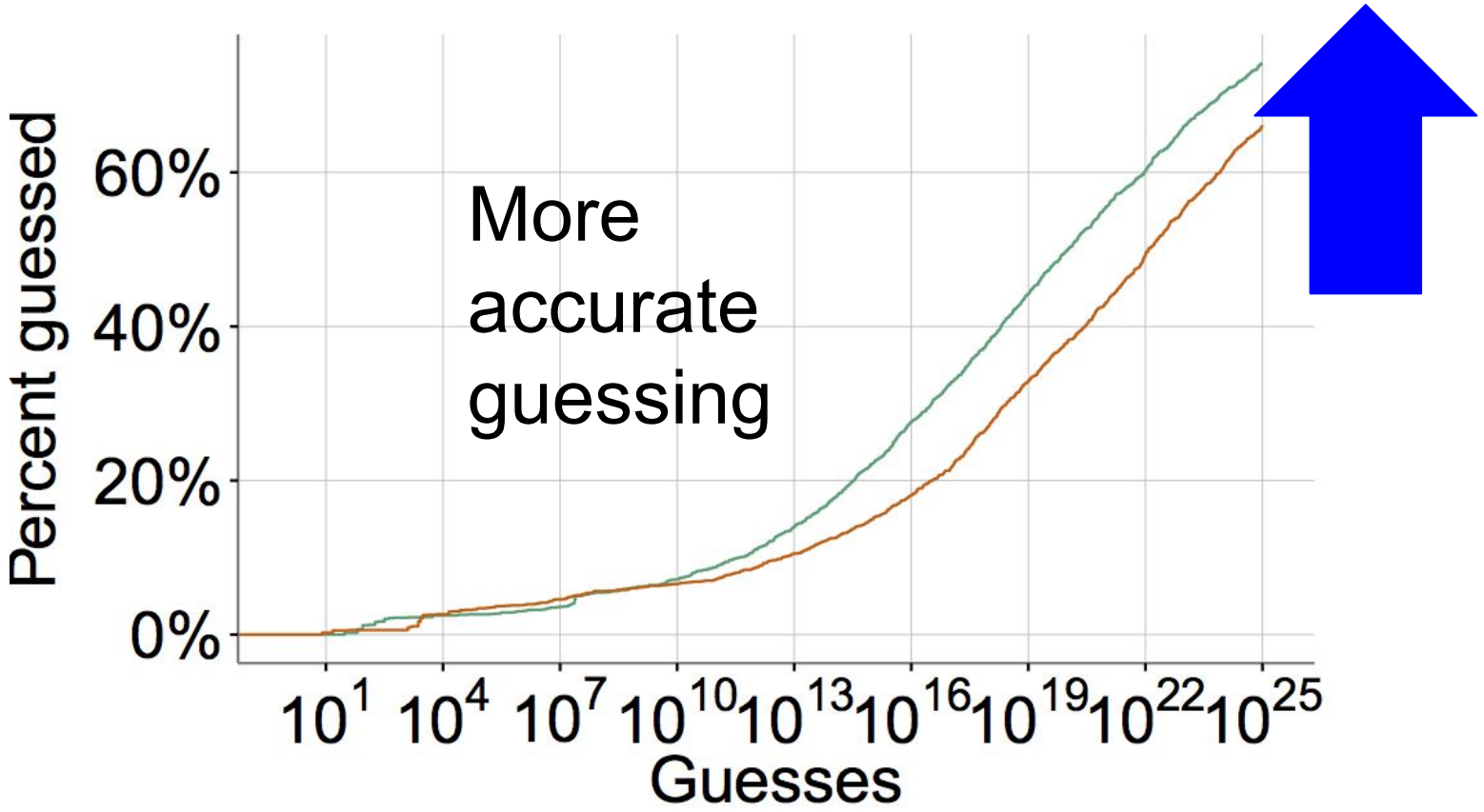
Comparison to other approaches


Percent guessed
60%
40%
20%
0%
10^1 10^4 10^7 10^{10} 10^{13} 10^{16} 10^{19} 10^{22} 10^{25}
Guesses

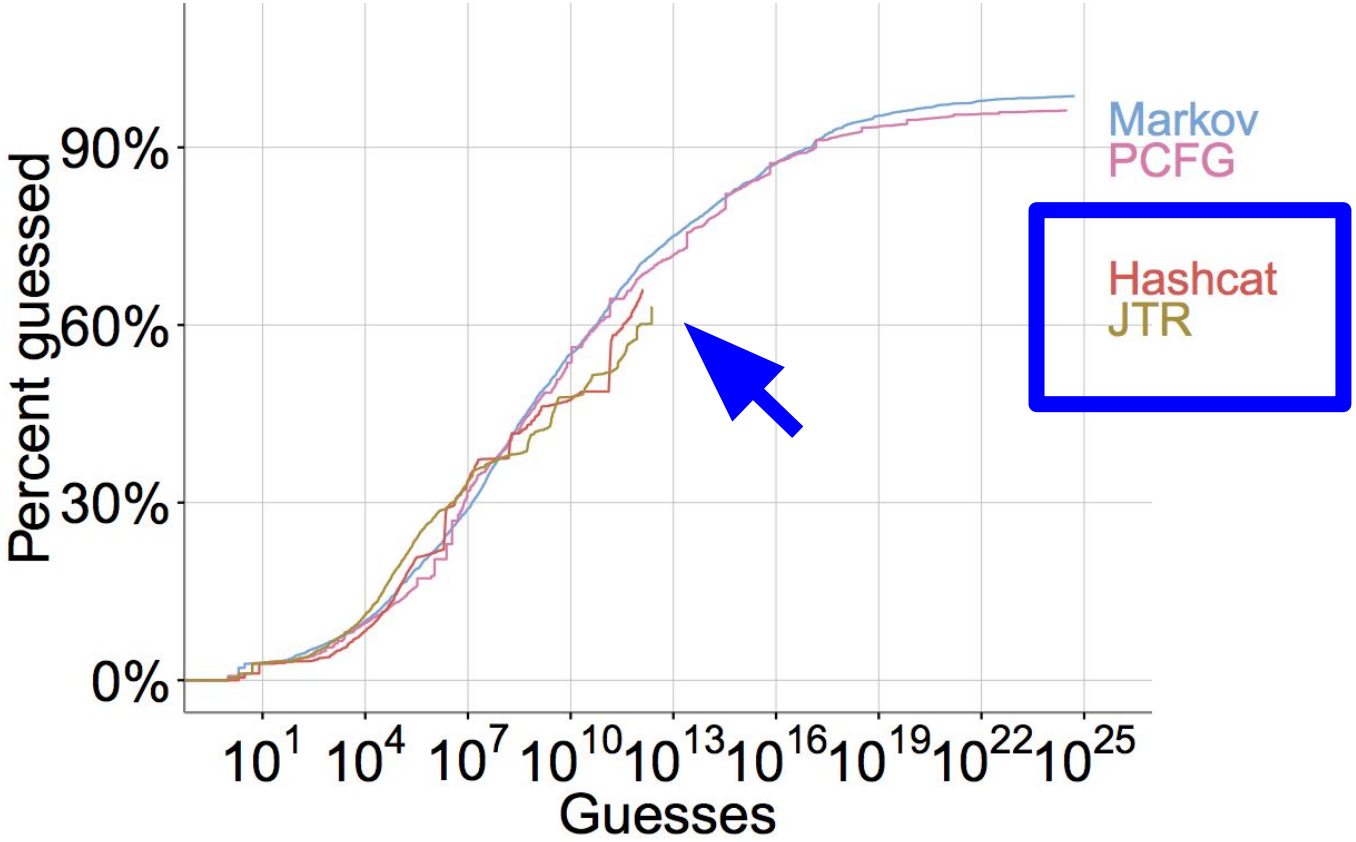

Percent guessed
60%
40%
20%
0%
10^1 10^4 10^7 10^{10} 10^{13} 10^{16} 10^{19} 10^{22} 10^{25}
Guesses


Percent guessed
60%
40%
20%
0%
10^1 10^4 10^7 10^{10} 10^{13} 10^{16} 10^{19} 10^{22} 10^{25}
Guesses

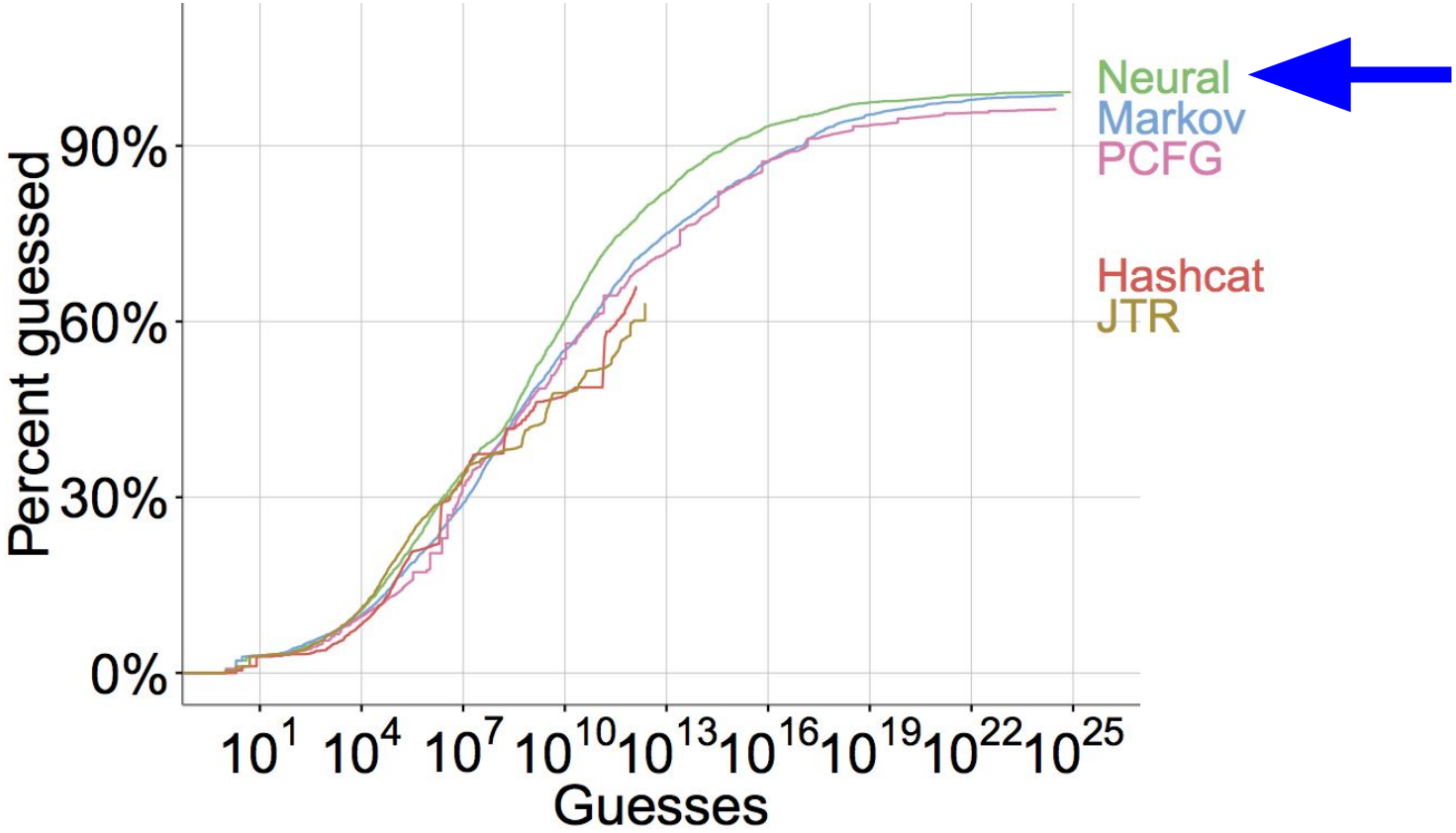

Percent guessed
60%
40%
20%
0%
More
accurate
guessing
10^1 10^4 10^7 10^{10} 10^{13} 10^{16} 10^{19} 10^{22} 10^{25}
Guesses


Percent guessed
60%
40%
20%
0%
More accurate guessing
10^1 10^4 10^7 10^{10} 10^{13} 10^{16} 10^{19} 10^{22} 10^{25}
Guesses

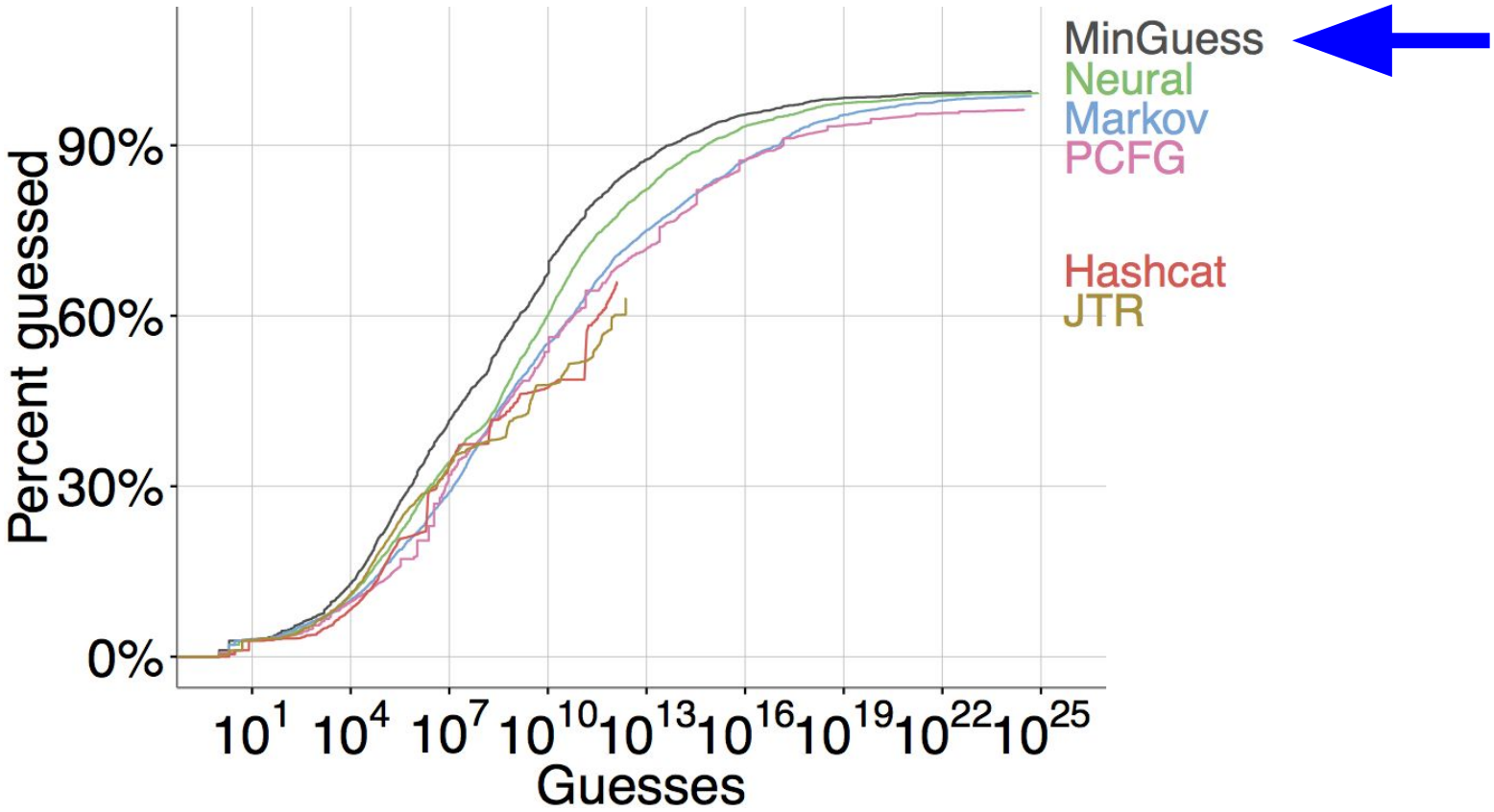
1class8: Comparison



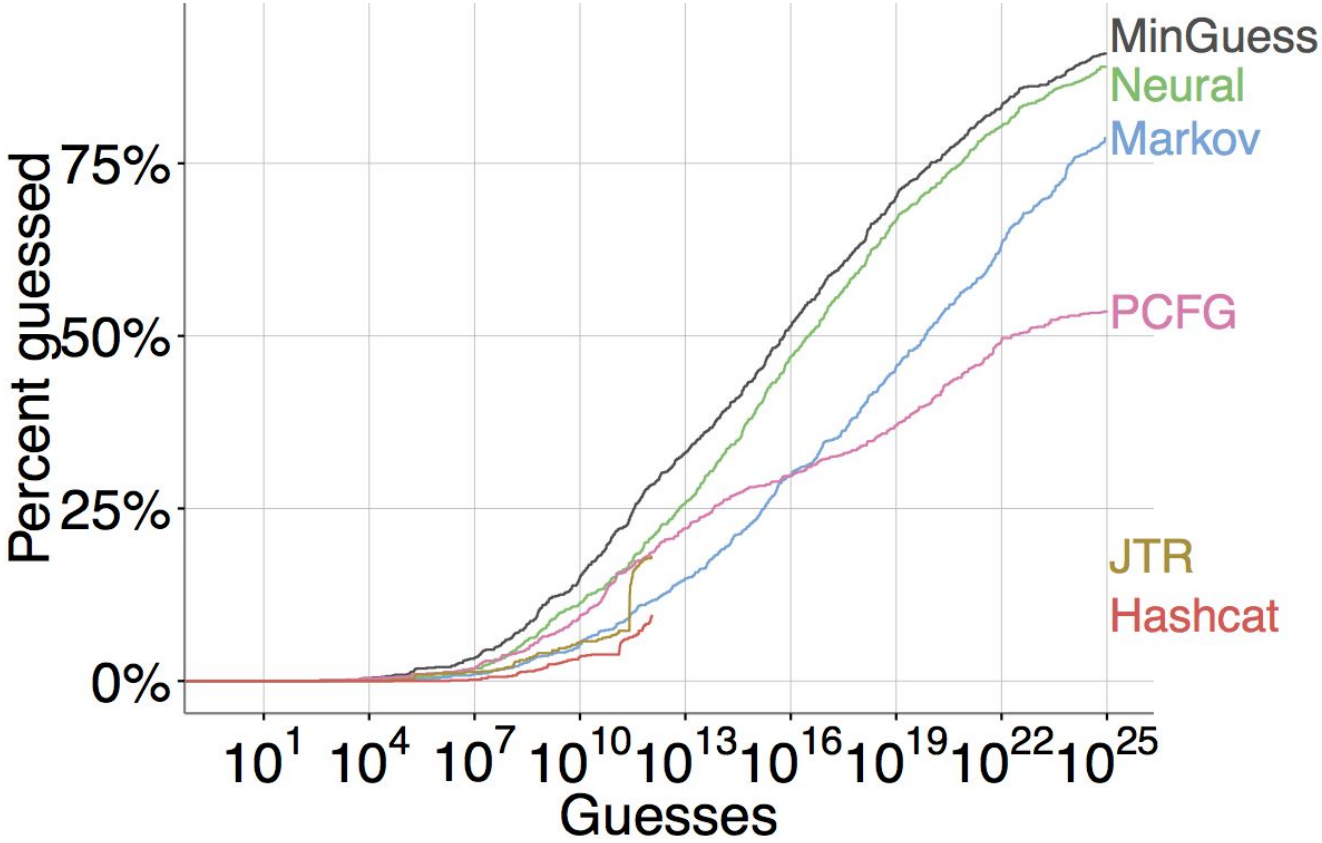
1class8: Neural Networks Guess Better



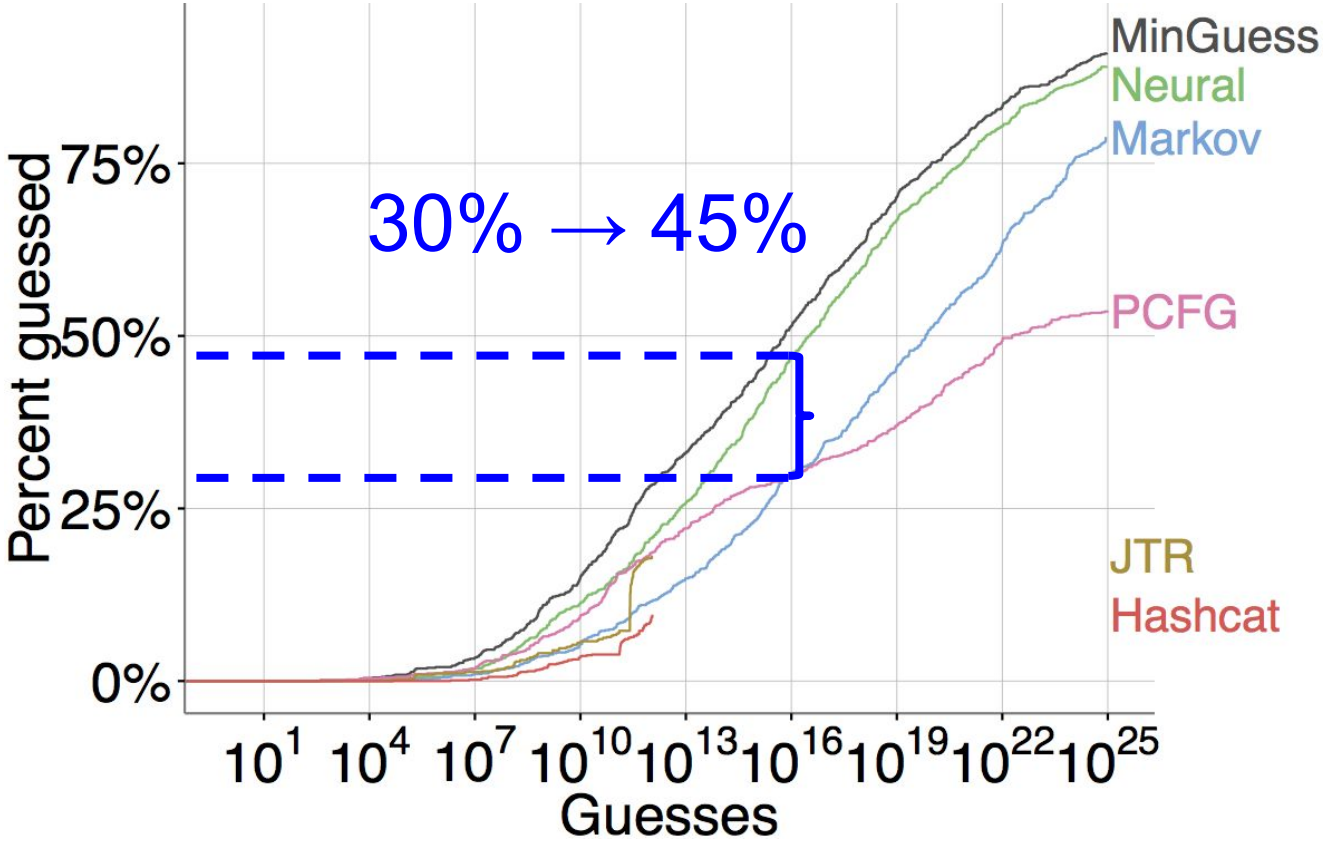
1class8: Neural Networks Guess Better



3class12: Neural Networks Guess Better



3class12: Neural Networks Guess Better



Outline: Guessing with Neural Networks

- Password guesser design
- Comparison to other methods
- **Real-time, in-browser feedback**

Current password feedback:

Quick ***or*** accurate

Accurate Guessing Methods

100s MB to GBs!

Accurate Guessing Methods

100s MB to GBs!

Accurate Guessing Methods

100s MB to GBs!

Neural networks: 60MB, 3MB

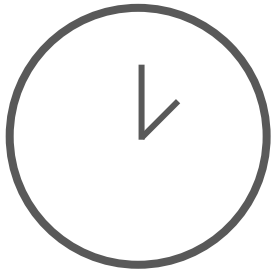
Accurate Guessing Methods

?

Neural networks: 60MB, 3MB

Accurate Guessing Methods

Hours to days!



Can neural networks give real-time feedback?

Ideal Model Targets

- Small: < 1MB
- Fast: < 0.1 sec
- JavaScript
- Accurate

Making Model Small

- Small version of neural network
- Quantize parameters of model
- Lossless compression

850KB < 1MB

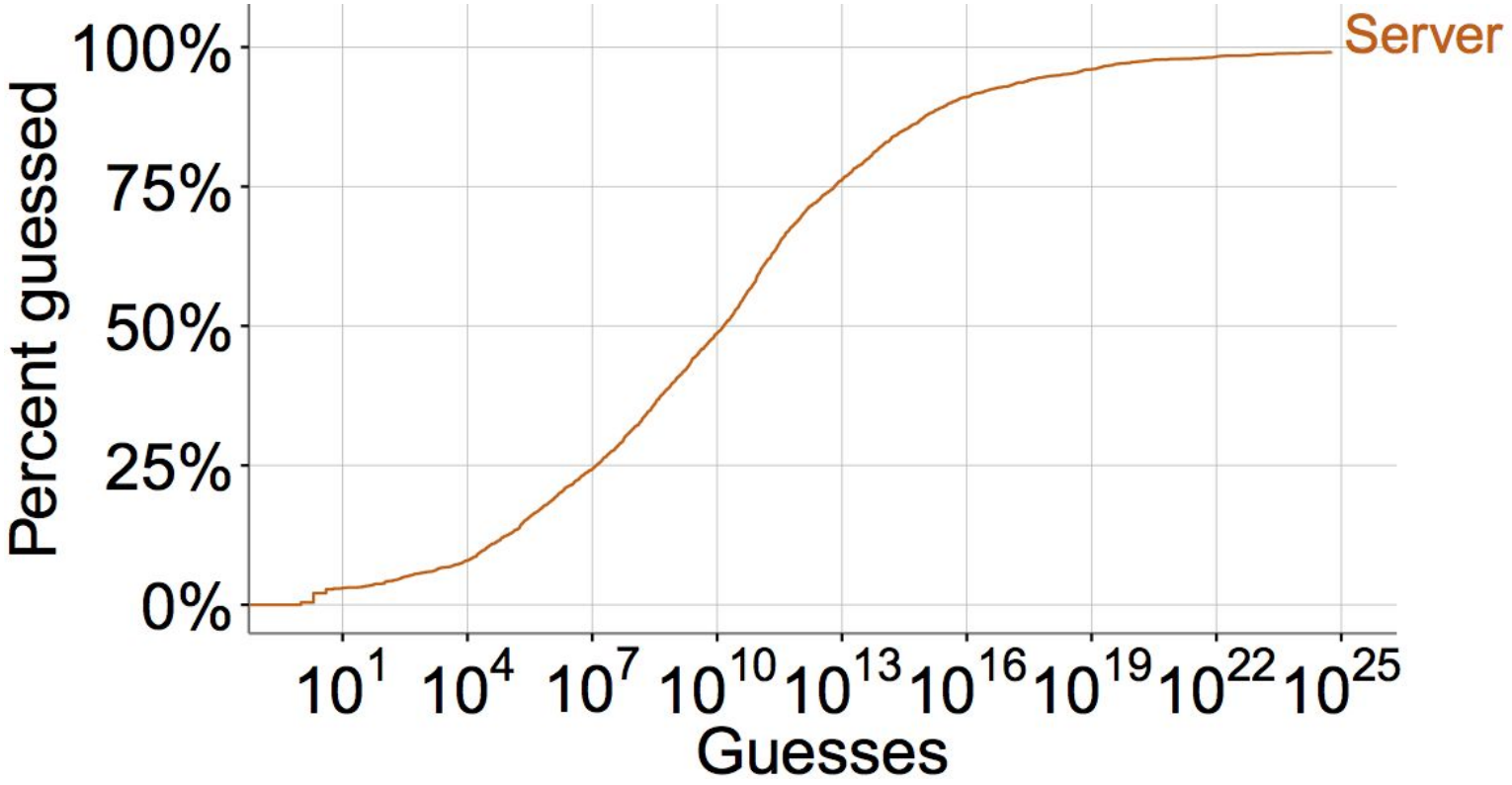
Making Model Fast

- Pre-compute inexact mapping from prob to guess number
- Cache intermediate results
- Run on separate thread

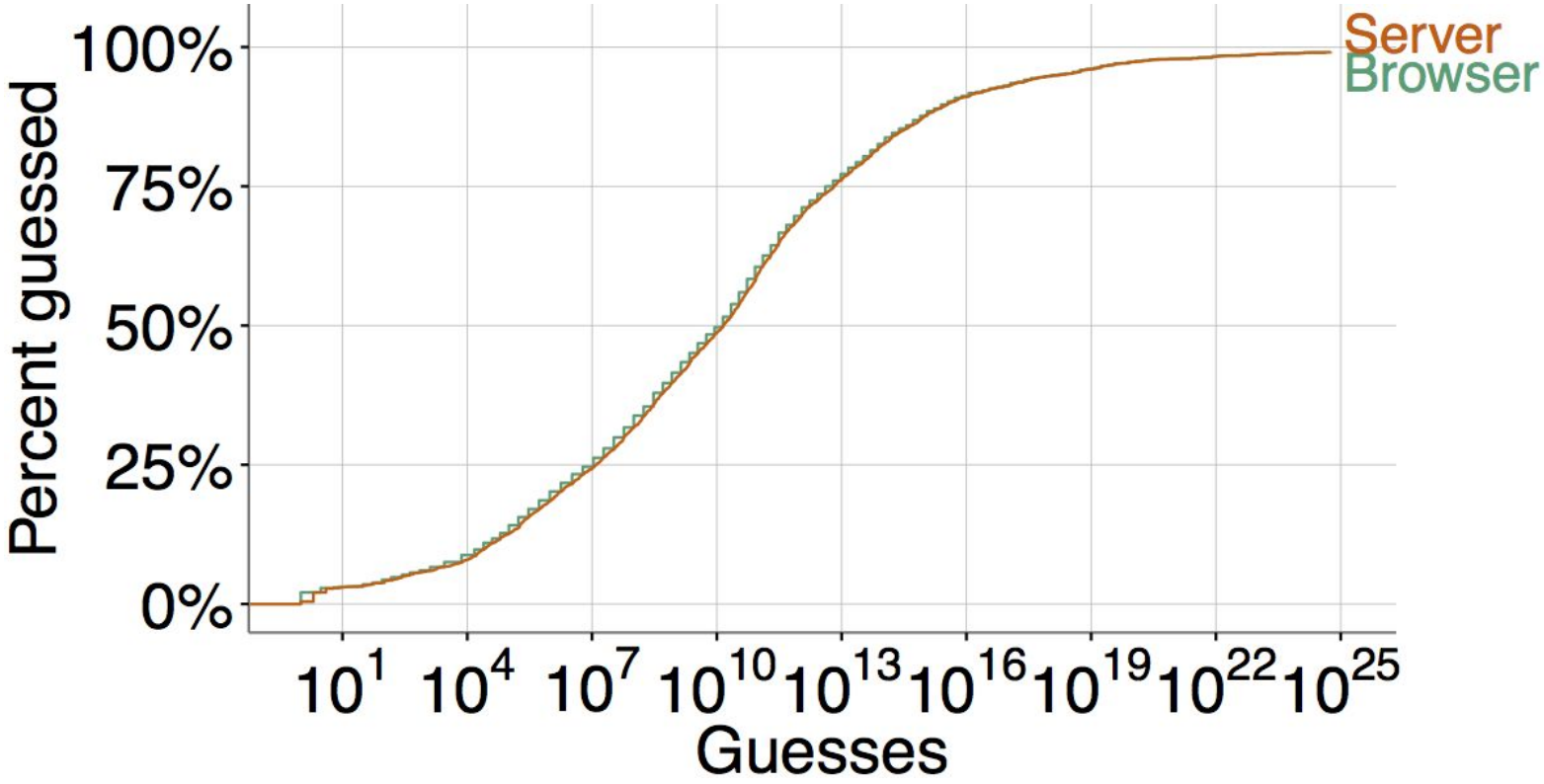
17 ms < 0.1 sec

How Accurate Is the Small, Fast Model?

How Accurate Is the Small, Fast Model?



How Accurate Is the Small, Fast Model?

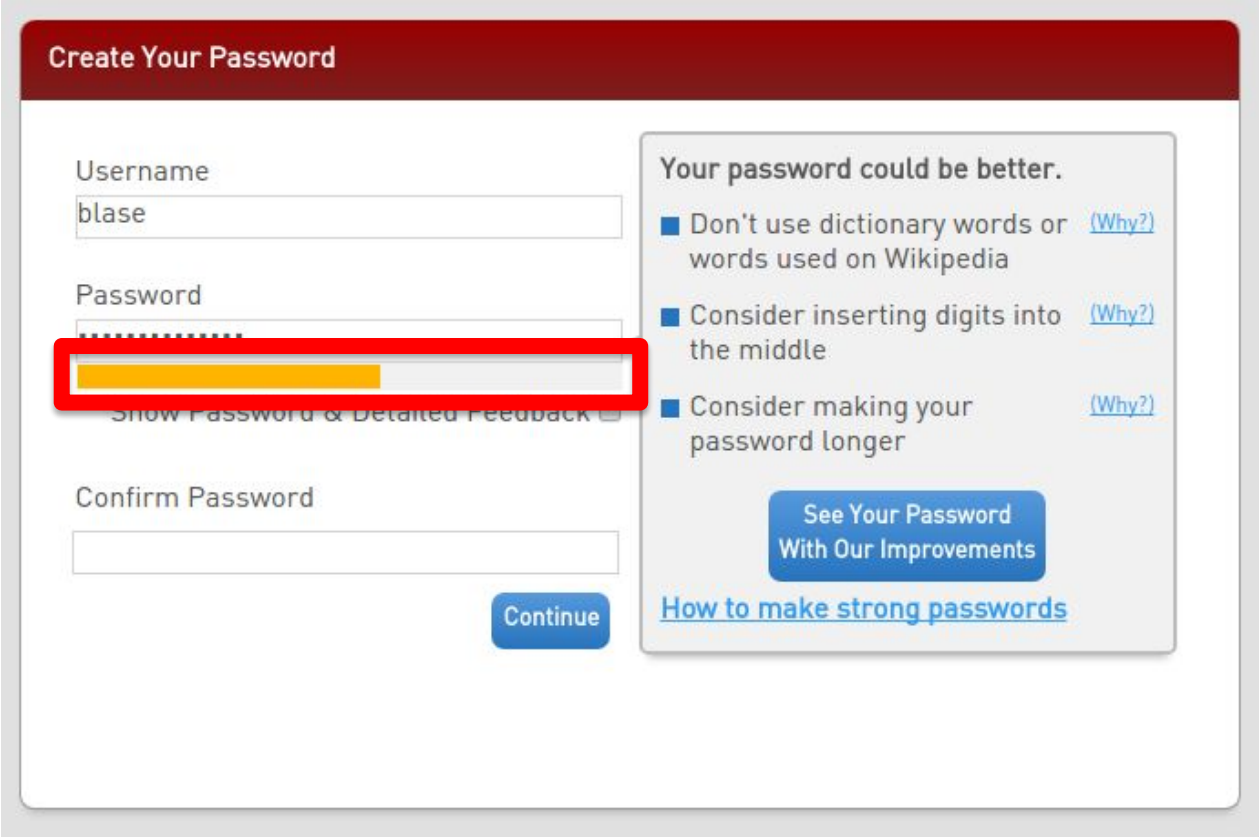


How Accurate Is the Small, Fast Model?

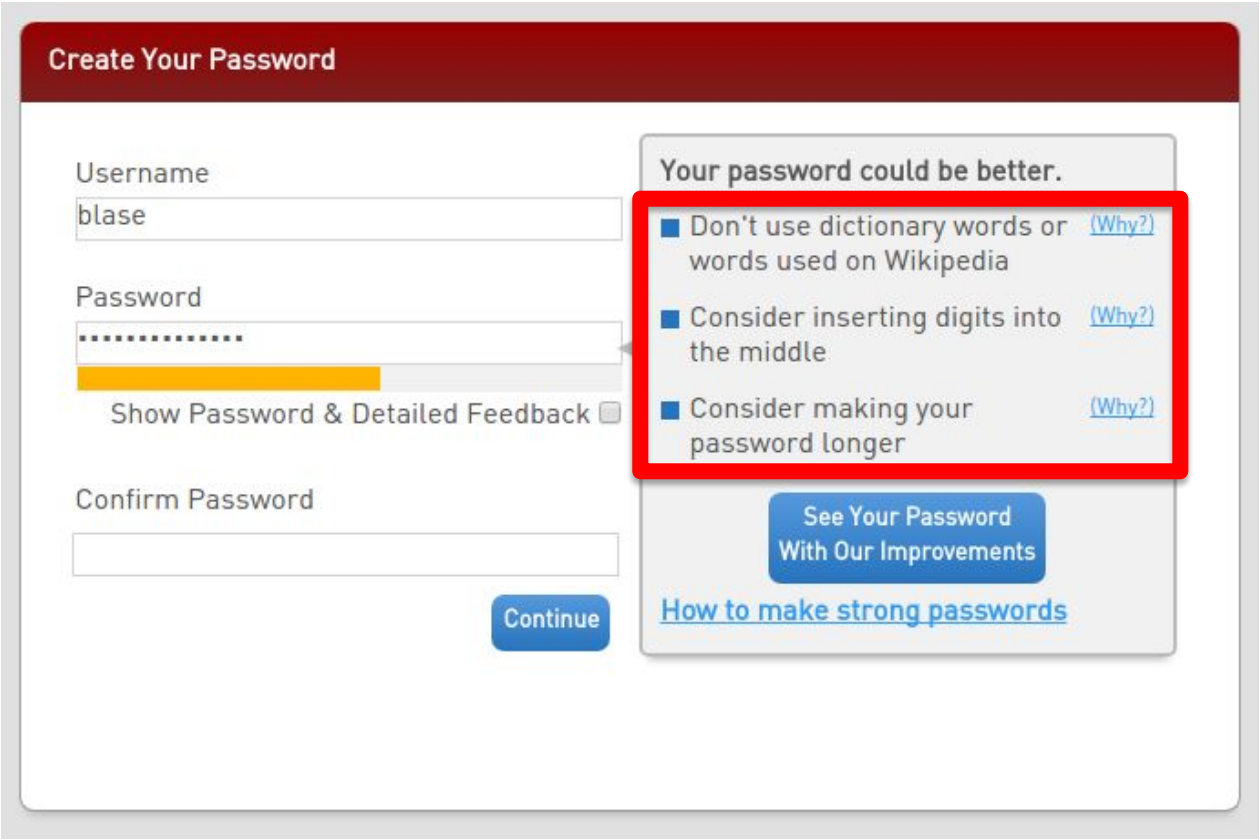
Does Measuring Password Strength Help?

[Design and Evaluation of a Data-Driven Password Meter
B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. Cranor, H. Dixon, P. Emami Naeini, H. Habib, N. Johnson, and W. Melicher. CHI'17]

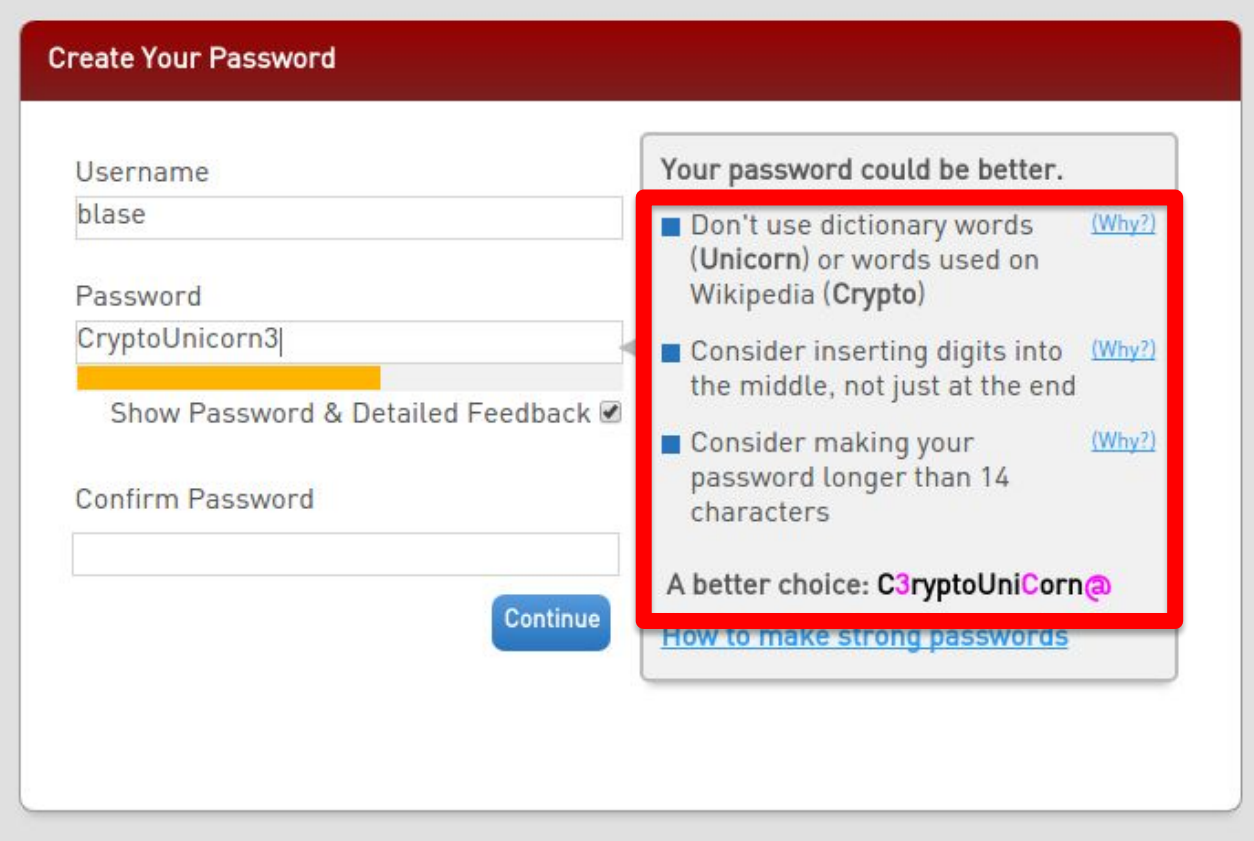
We Developed and Tested a Meter GUI



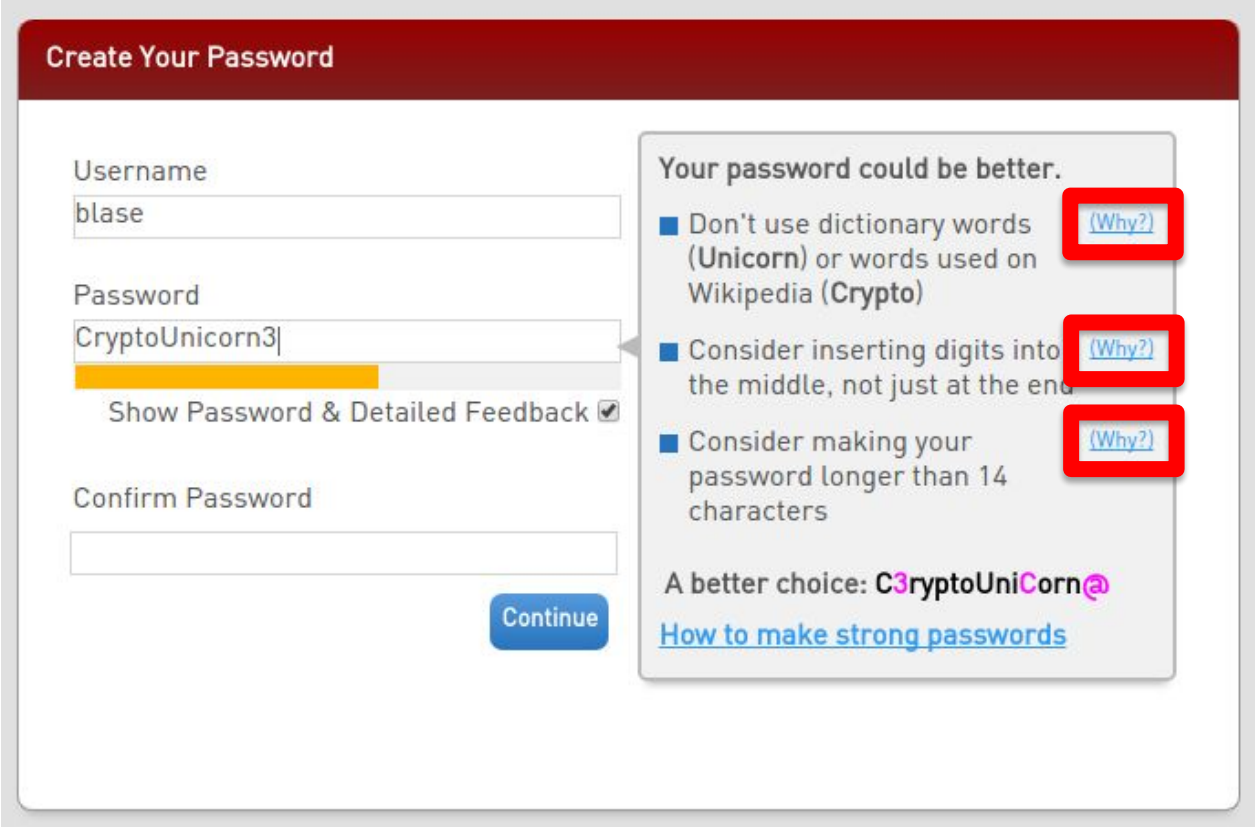
Provides Text Feedback



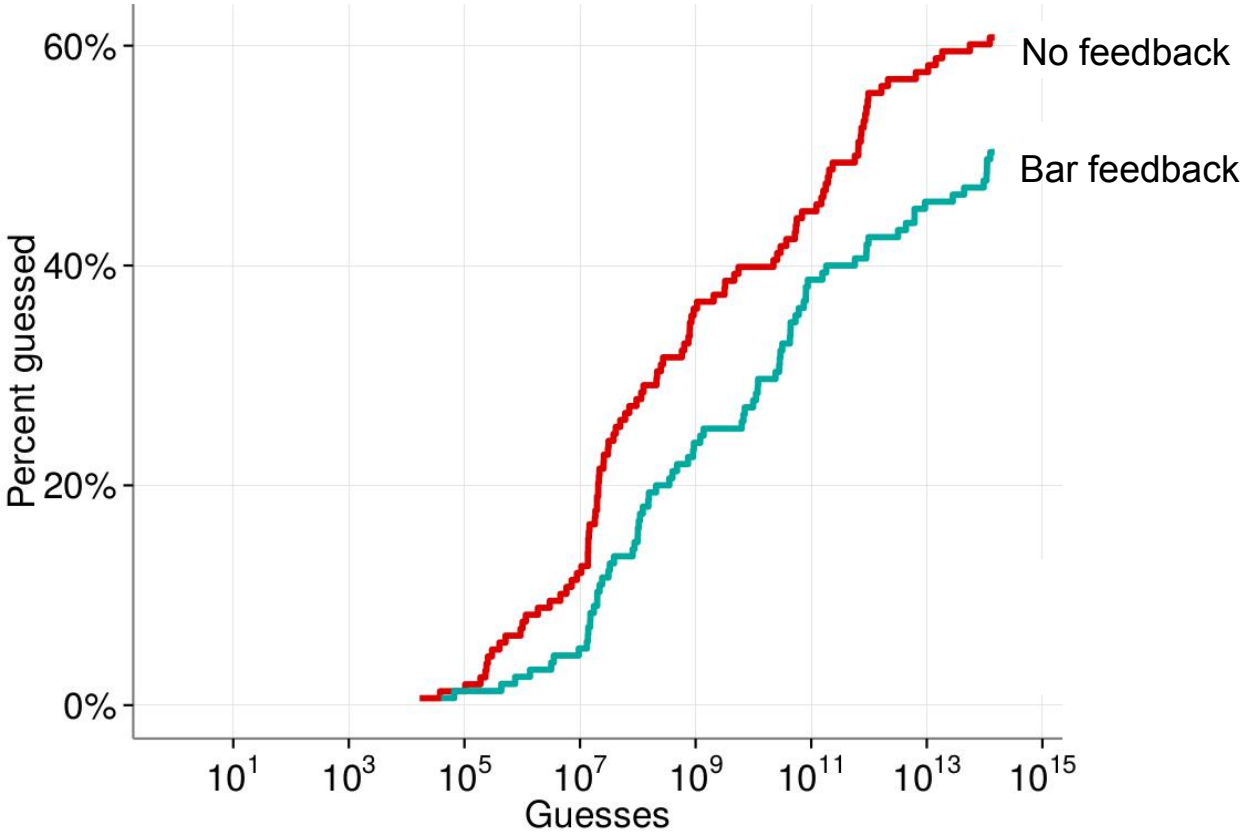
Gives Detail (Password Shown)



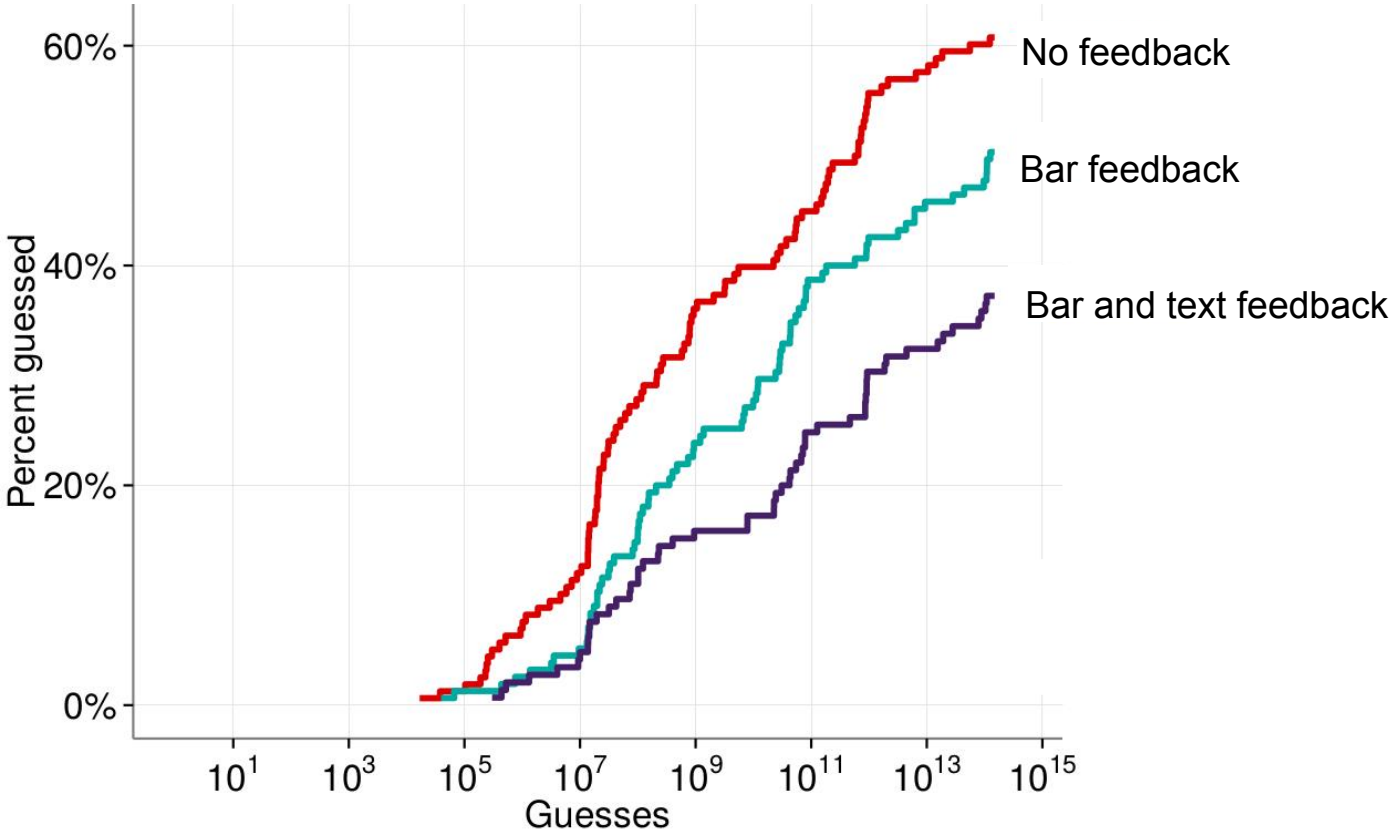
Offers Explanations



Does Measuring Strength Help? Yes!



Does Measuring Strength Help? Yes!



Modeling Passwords Using Neural Networks

- Neural networks guess passwords accurately
- Can be made small and fast for client-side feedback

`github.com/cupslab`

William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor

Carnegie Mellon University