MoLe: Motion Leaks through Smartwatch Sensors

Yi Mao, Ruoxi Li

Introduction

Question: can we mine accelerometer and gyroscope data from smart watches to infer what words a user is typing?

Introduction - Challenges

- Absence of data from the right hand (which is not wearing the watch)
- Issue of inferring which finger executed the key-press
- For a given watch position, a shortlist of keys could have been pressed
- Different users’ habits and keyboard devices

Introduction - Opportunities

- the watch motion is mostly confined to the 2D keyboard plane
- The orientation of the watch is relatively uniform across various users
- knowing spelling priors from English dictionary further helps in developing Bayesian decisions.

Introduction - Work

Motion Leaks (MoLe)

- Device: Samsung Gear Live smartwatch
- Input:
  - 2 authors (attacker) - each typed 500 words for training
  - 8 volunteers (attackee) - each typed 300 words for testing
- Output: K words (ranked in decreasing probability) short-list for each word

Introduction - Contributions

- Identifying the possibility of leakage - required building blocks
  - key-press detection
  - hand-motion tracking
  - cross-user data matching
  - Bayesian inference
- Developing the system on Samsung Gear Live
  - experimenting with real users
  - reasonable accuracy
  - Sense of alarm

A first look at smart watch data

A first look at smart watch data

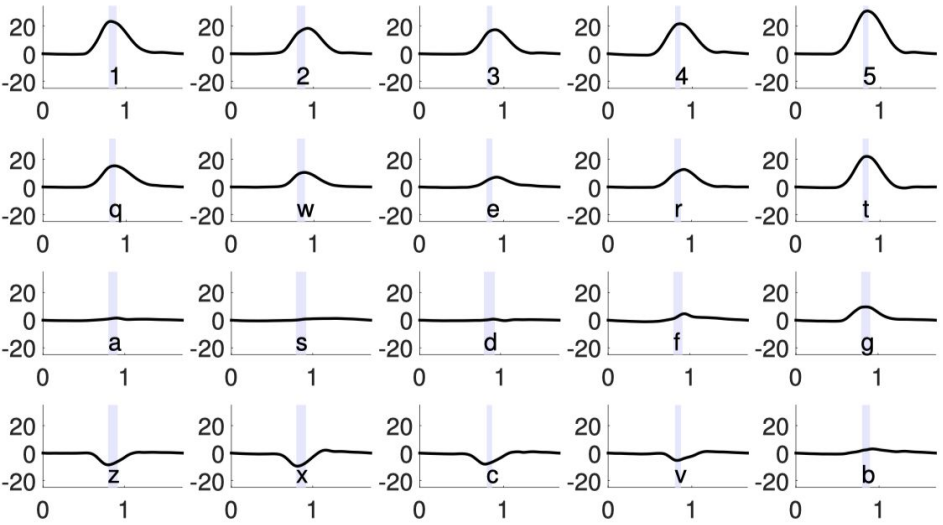


Figure 3: The watch X axis displacements while a human types 20 characters. In the figures, X axis is time in seconds and Y axis is watch X axis displacement in millimeter. The gray bar shows the keystroke press and release time interval.

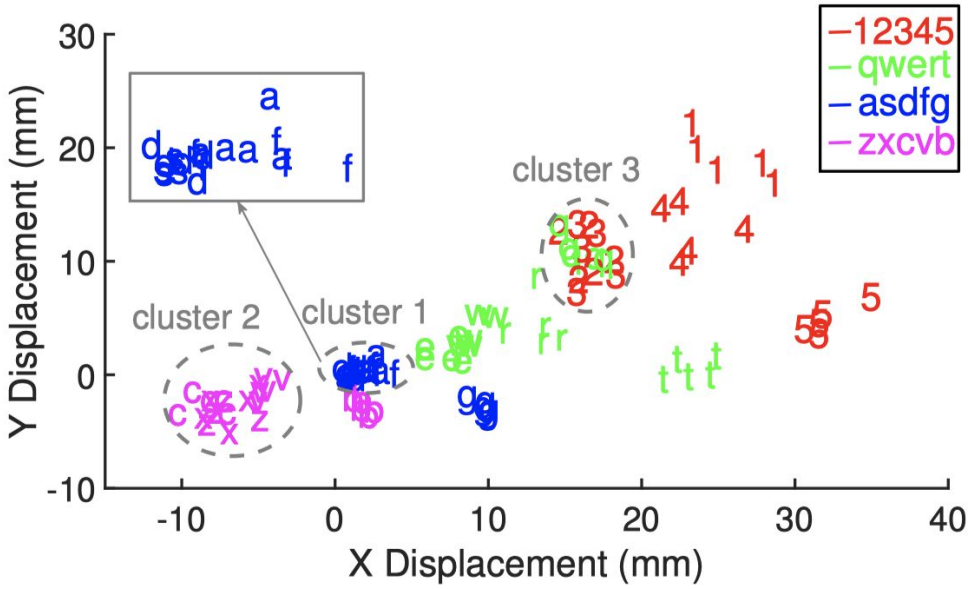


Figure 4: Watch 2D displacements while a human types 20 characters with her left hand. Each character is typed repeatedly 5 times. (0,0) is the initial location when left hand fingers are placed on home position ("asdf"). Note that X and Y axes in the graph are in the watch's coordinate system.

A first look at smart watch data

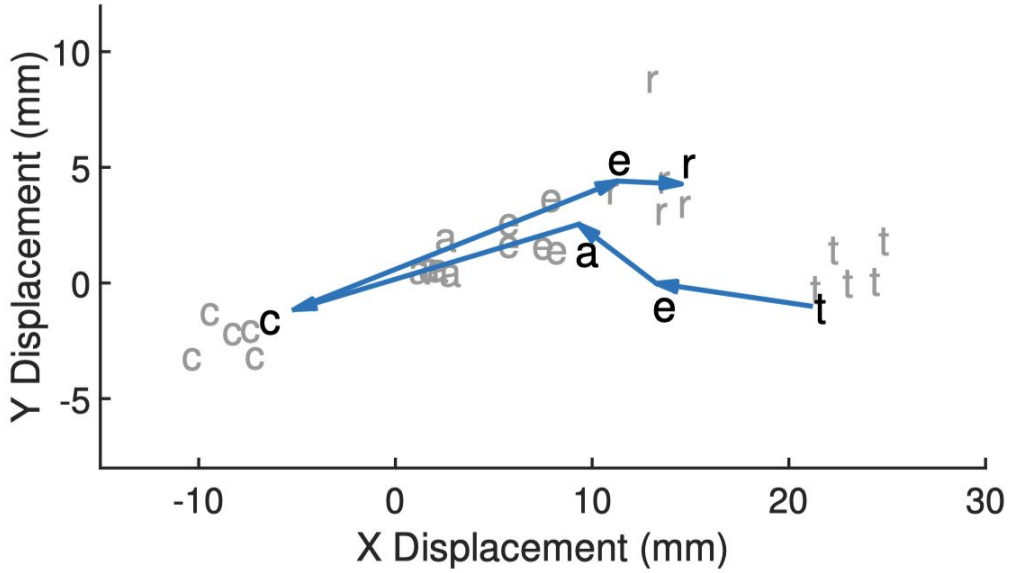


Figure 5: Comparison of typing "teacher" continuously (in black) against each character separately (in gray). Note that the positions of "e", "a" and "c" are away from their original points due to sequential typing. Also, "h" is not captured due to right hand typing.

System Overview

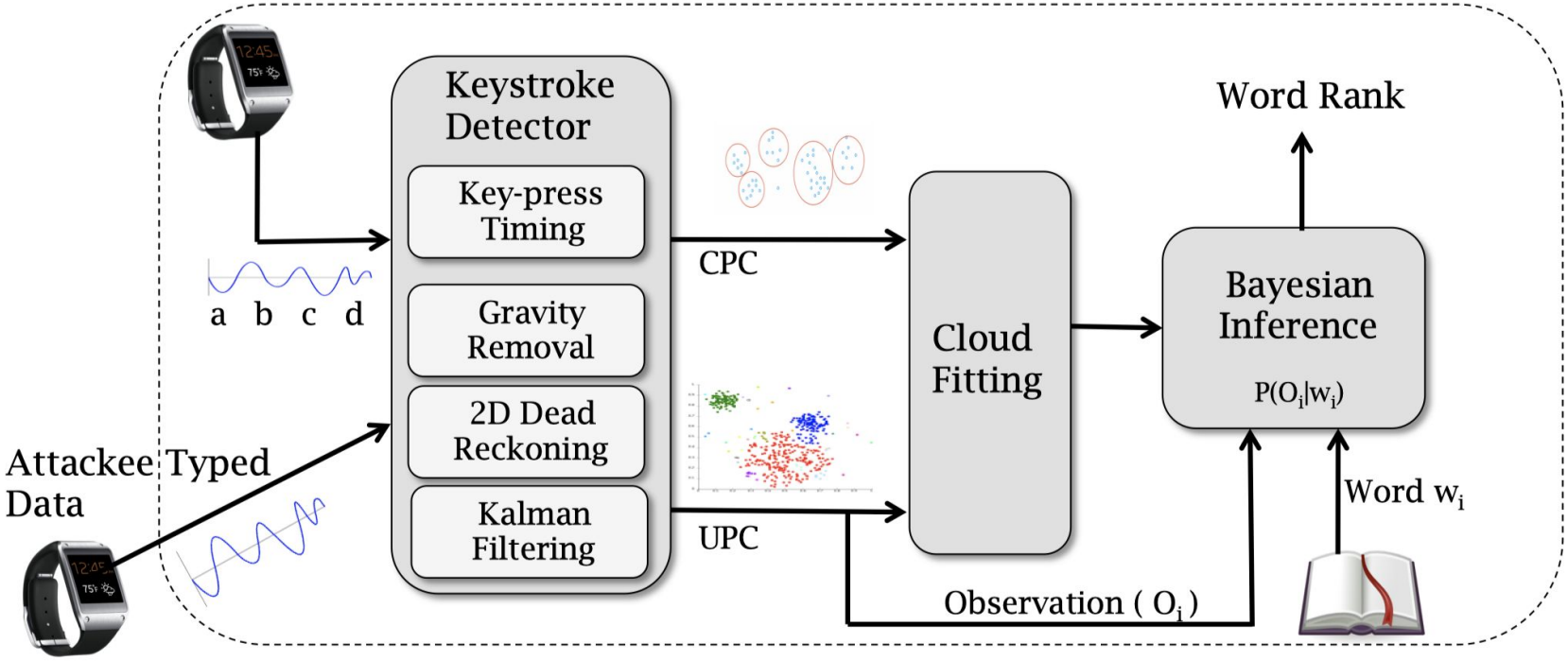


Figure 6: System Overview: The typed data from users are pre-processed through gravity removal and timing analysis blocks, superimposed on the refitted typing templates, and passed through a Bayesian inference model that leverages the patterns and structures in English words to ultimately decode the typed words. Note, training is only required from the attacker's end; no training needed for the user.

System Overview

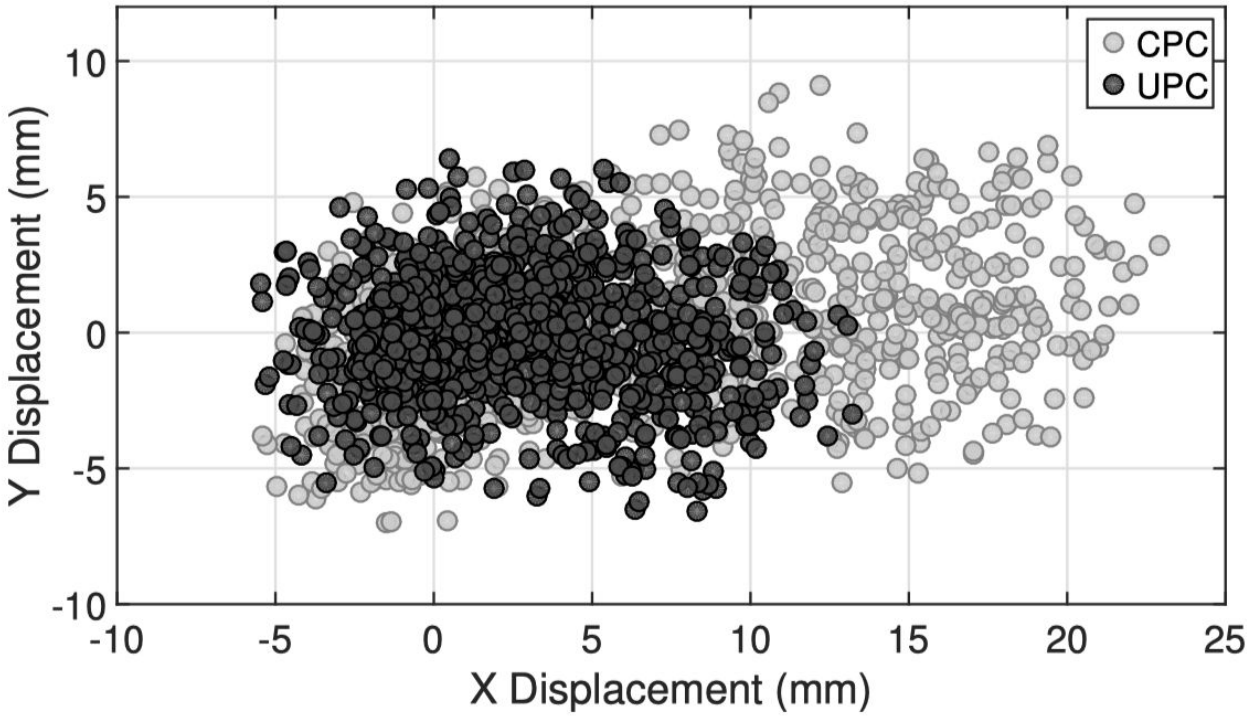


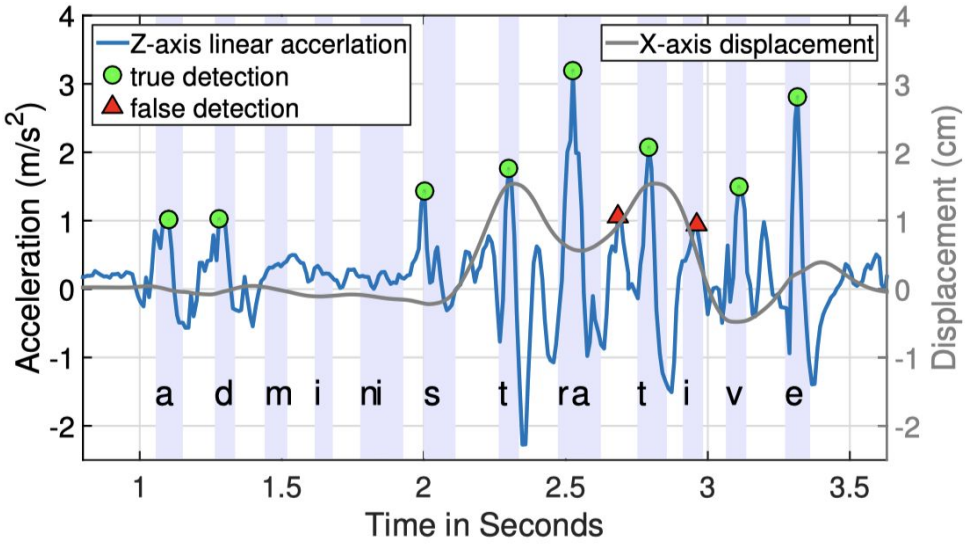
Figure 7: (a) Character point cloud computed from attackers data; (b) Unlabeled point cloud computed from user's data.

System Overview - Assumptions

- The evaluation is performed in a controlled environment where volunteers type **one word at a time** (as opposed to free-flowing sentences).
- We assume **valid English words** – passwords that contain interspersed digits, or non-English character-sequences, are not decodable as of now.
- We have used the **same** Samsung **smart watch model** for both the attacker and the user – in reality the attacker can generate the CPC for different watch models and use the appropriate one based on the user's model.
- We assume the **user** is seasoned in **typing** in that he/she roughly **uses the appropriate fingers** – novice typists who do not abide by basic typing rules may not be subject to our proposed attacks.

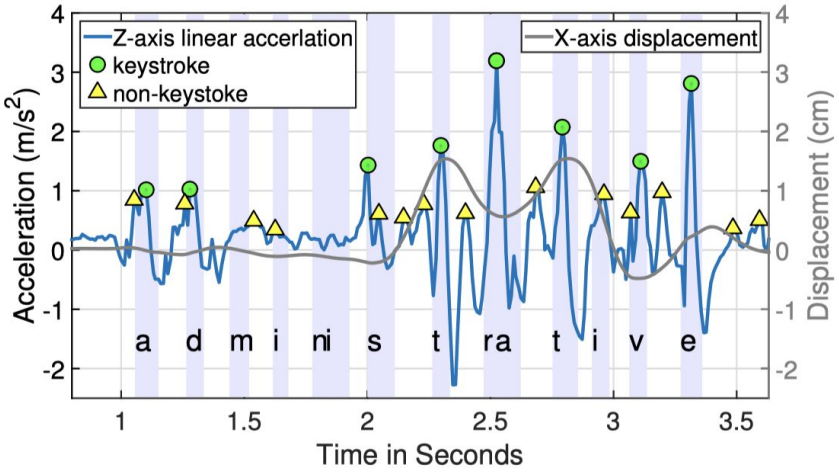
Keystroke Detector - Keypress timing

- Intuition - key presses is rooted in the hand's motion in the vertical direction
  - Peak motion in Z axis of the watch
- False positive - peaks caused by hand movement during transition
- False negative - subtle motion for typing keys like "asdf"



Keystroke Detector - Keypress timing

- Simple peak detection + bagged decision trees
  - low threshold for peak detection
  - features for distinguishing keystrokes among peaks: the width, height, prominence of the Z axis peak; the mean, variance, max, min, skewness, kurtosis for each of the 3-axis displacement, velocity, acceleration, gyroscope rotation; the magnitude of acceleration/gyroscope; the correlation of each pair between acceleration, gyroscope vectors



Keystroke Detector - Keypress timing

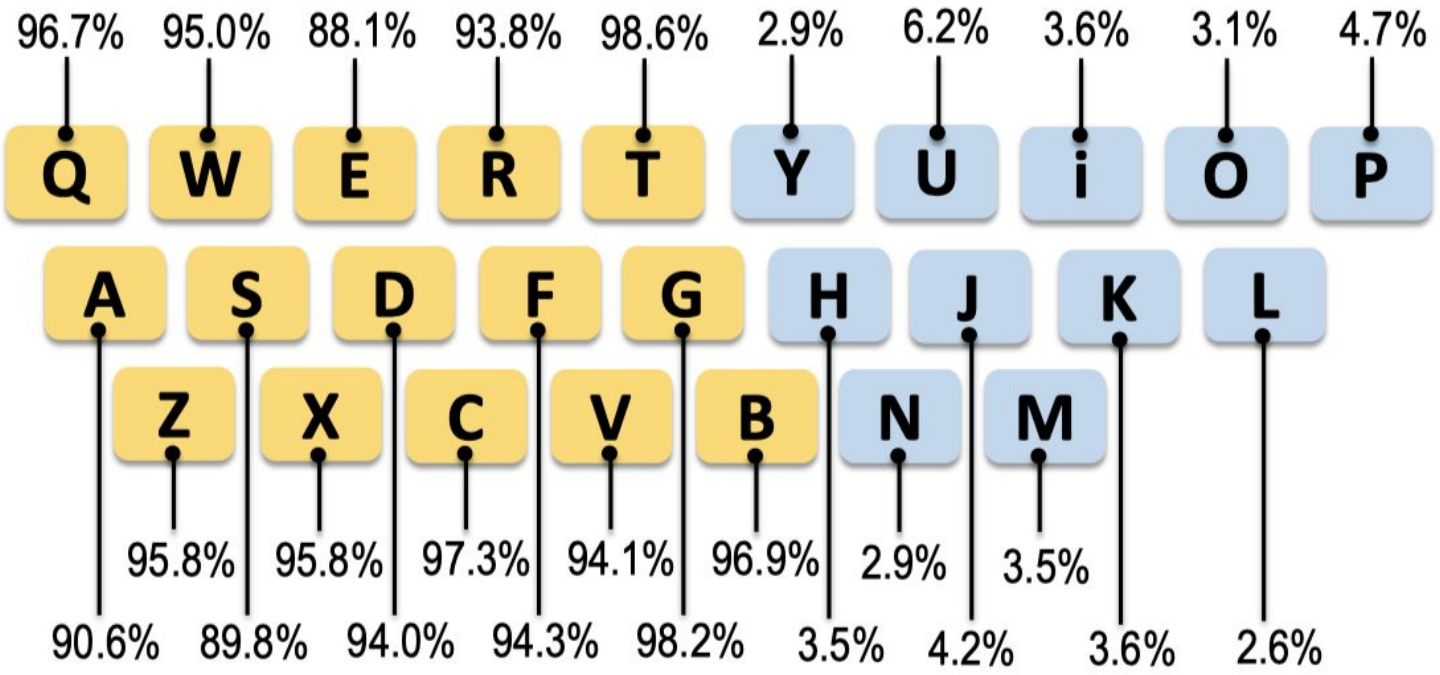
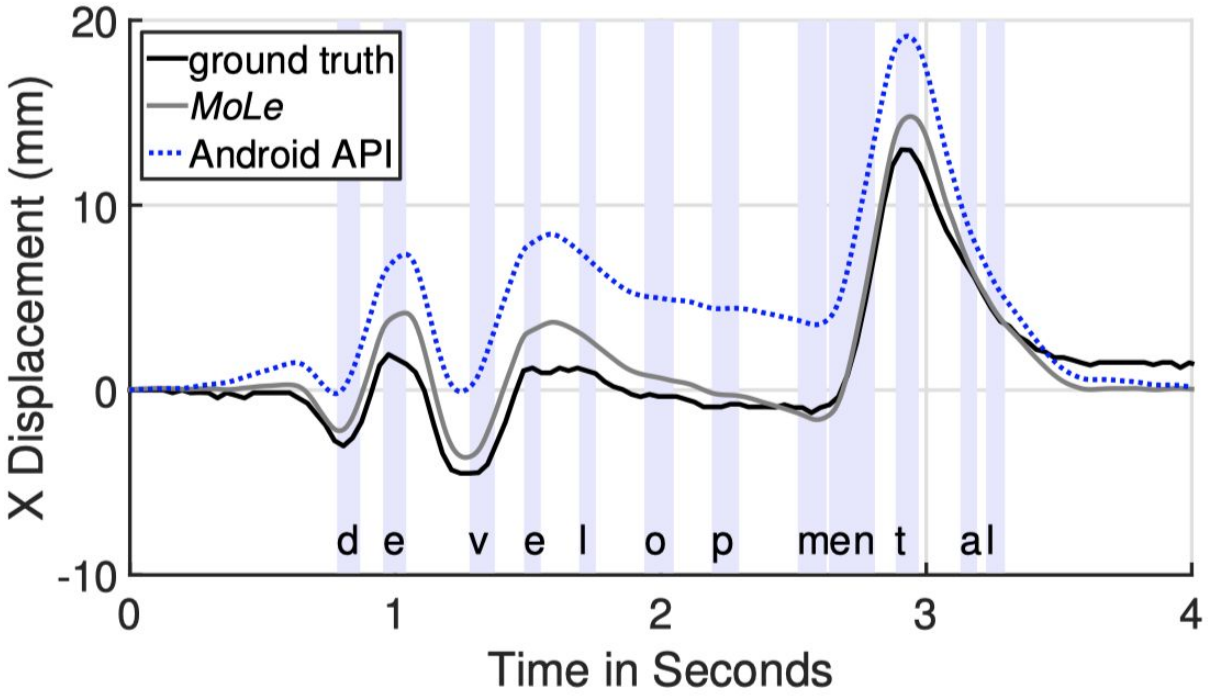


Figure 10: Keystroke detection rate for each character.

Keystroke Detector - Keypress Location Estimation

- Mole requires high accuracy, but native Android API is inadequate



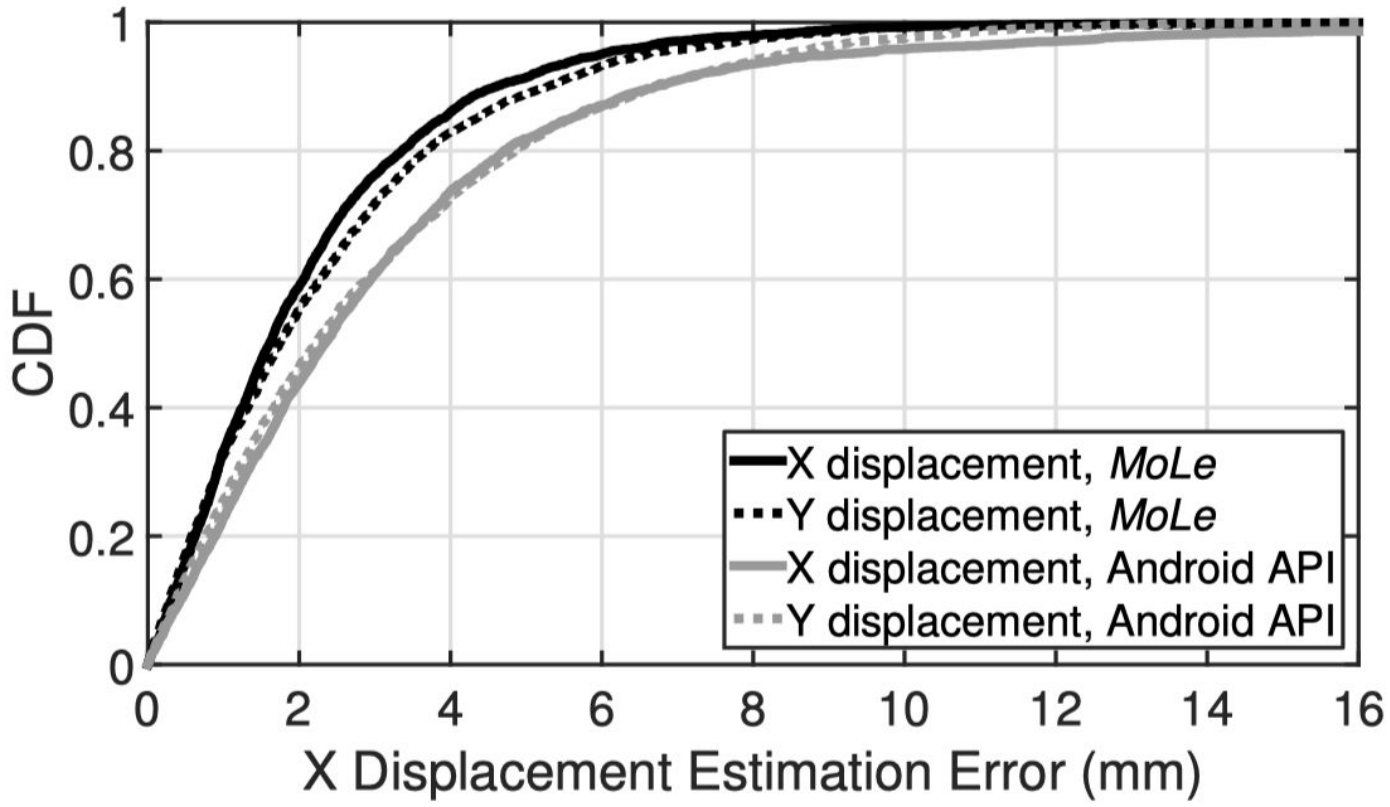
Keystroke Detector - Keypress Location Estimation

- Find gravity to define an absolute coordinate system.
  - gravity's direction can be determined before typing
  - absolute horizontal plane is orthogonal to gravity
  - convert watch's x-axis to absolute x-axis by projection, y-axis is then computed from cross product of x-axis and z-axis (gravity)
- Estimate and remove gravity.
  - Use gyroscope to estimate variation in gravity g(t)
  - arg(t) = a(t) - g(t) (in watch's coordinate system)

Keystroke Detector - Keypress Location Estimation

- Estimate C(t) and calculate projected acceleration.
  - integrate arg(t) directly doesn't make sense, as watch rotates overtime
  - project arg(t) to absolute coordinate system, then double integrate
- Calibrate by mean removal (speed and displacement).
  - errors accumulate when double integration
  - when watch stops, v(T) = 0 and s(T) = 0
- Kalman smoothing.
  - gravity estimation is not reliable, i.e. has an error $g'_e(t)$
  - think arg'(t) (measured) = $g'_e(t)$ + noise (true arg(t))
  - Compute $g'_e(t)$ with Kalman smoothing
  - arg(t) = arg'(t) - $g'_e(t)$

Keystroke Detector - Keypress Location Estimation



Point Cloud Fitting

- relative motions (relative locations of the point clouds) between keys should bear similarity across all users.
- the fitting parameters for up and down hand displacements can be different, therefore, 2 convex hulls for positive and negative displacement respectively

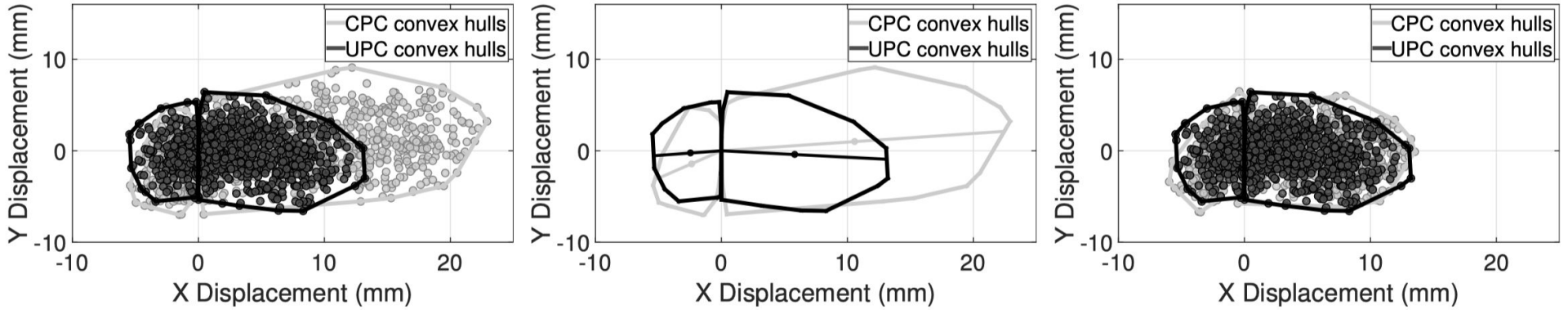


Figure 13: Point Cloud Fitting. Black points are the CPC attacker template and gray points are UPC from attackee. (a) Finding each convex hull (b) Calculate the centroids and perform rotate and scale. (c) Point cloud fitting result.

Bayesian Inference

$$P(W \mid O) = \frac{P(O \mid W) P(W)}{P(O)} \qquad P(W \mid O) \propto P(O \mid W)$$

- *W* is a candidate word from the dictionary and *O* is the observation motion data
- *P*(*W*|*O*) is the posterior probability of the word given the observed motion data
- *P*(*O*|*W*) is the likelihood function that estimates the probability of the word *W* based on the observed motion data
- *P* (*W*) is the prior probability which captures the word's occurrence frequency
- *P*(*O*) is the probability of the observation

Bayesian Inference - Number of Keystrokes

$$P(O \mid W) = P(N \mid W) = \sum_{(\alpha_1,\ldots,\alpha_N)} P((c_{\alpha_1},\ldots,c_{\alpha_N}) \mid W)$$

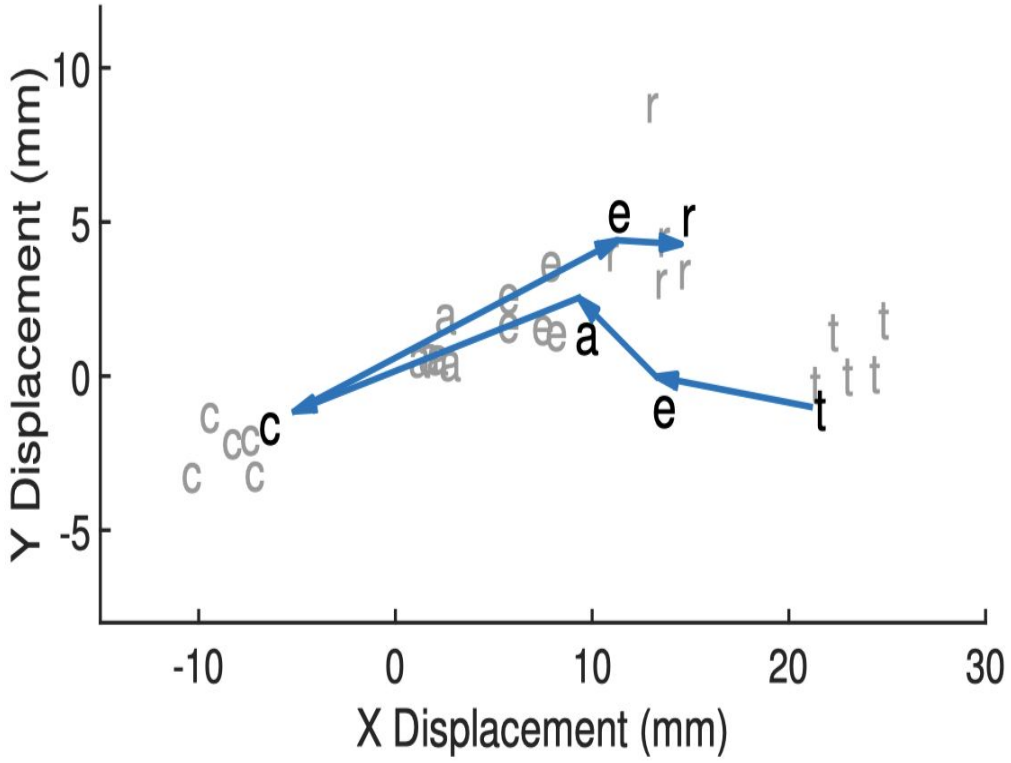
- *N* is the number of keystroke
- (α1,...,α*N*) represents one possible *N*-element subset of {1, 2, ..., *L*} (L is the word length)
- *P*((*c*α1 ,...,*c*α*N*) | *W*) is the probability that *N* keystrokes are generated by *c*α1 , ..., *c*α*N* .

Bayesian Inference - Watch Displacement

$$P(O \mid W) = \sum_{(\alpha_1,\ldots,\alpha_N)} P((c_{\alpha_1},\ldots,c_{\alpha_N}) \mid W) \prod_{i=1}^{N} p(d_i \mid c_{\alpha_i})$$

p(*di* | *cαi*) is probability density of *di* given character *cαi* .

Bayesian Inference - Character Transitions



$$P(O \mid W) \approx \sum_{(\alpha_1,\ldots,\alpha_N)} P((c_{\alpha_1},\ldots,c_{\alpha_N}) \mid W) \prod_{i=1}^{N} p(d_i \mid c_{\alpha_i}, c_{\alpha_{i-1}})$$

Bayesian Inference - Keystroke Interval

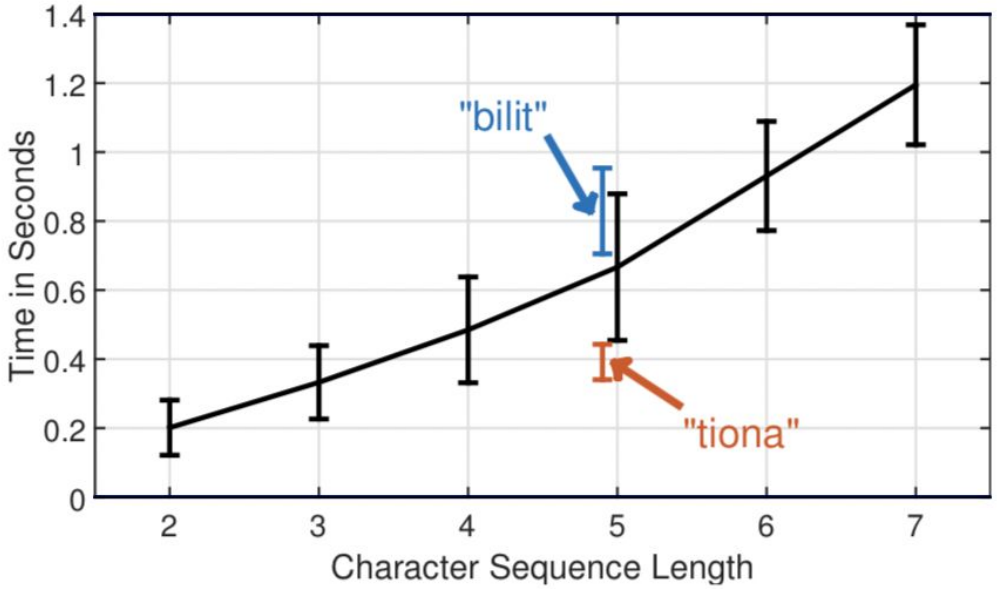


Figure 16: Y-axis is the time interval between two detected keystrokes and X-axis is the number of sequence length.

$$
\begin{aligned}
& P(O \mid W) \\
& = P(N \cap d_i, i = 1,2,...,N \cap t_j, j = 1,2,...,N-1 \mid W) \\
& \approx \sum_{(\alpha_1,...,\alpha_N)} P((c_{\alpha_1},...,c_{\alpha_N}) \mid W) \prod_{i=1}^{N} p(d_i \mid c_{\alpha_i}, c_{\alpha_{i-1}}) p((t_1,...,t_{N-1}) \\
& \mid (c_{\alpha_1},...,c_{\alpha_N}), (d_1,...,d_N), W)
\end{aligned}
$$

t h a n k s

(characters Stroke by left hand)

Performance - How good is *MoLe*?

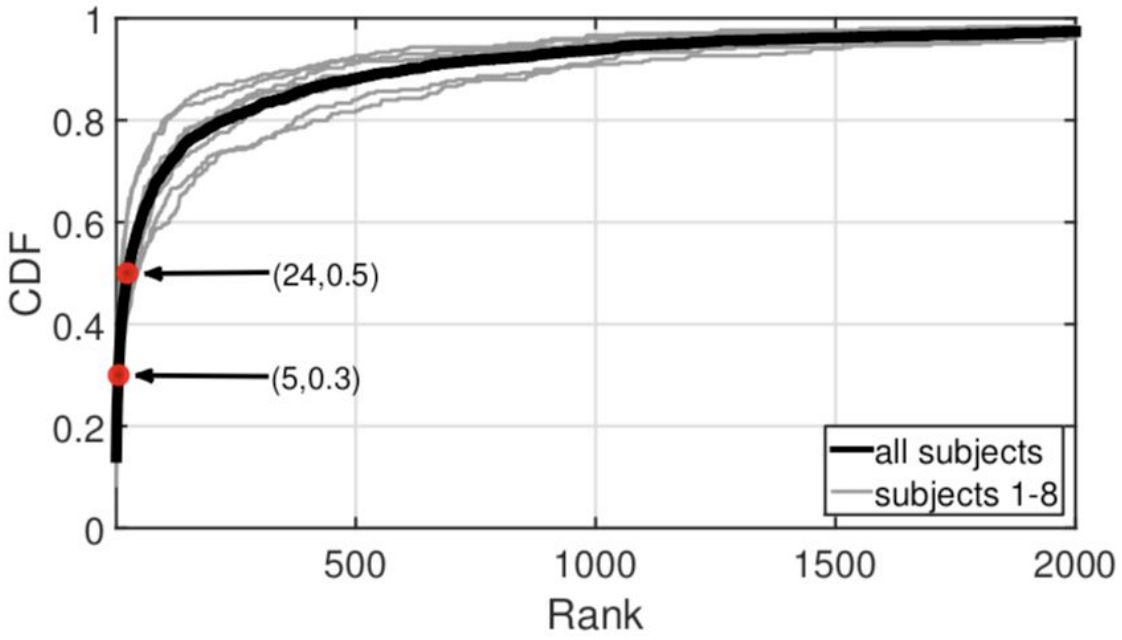


Figure 17: CDF of rank computed for each of the 2400 words typed by 8 subjects.

The median rank of a word is 24

While for 30 percentile, the rank is 5.

Performance - How good is *MoLe*?

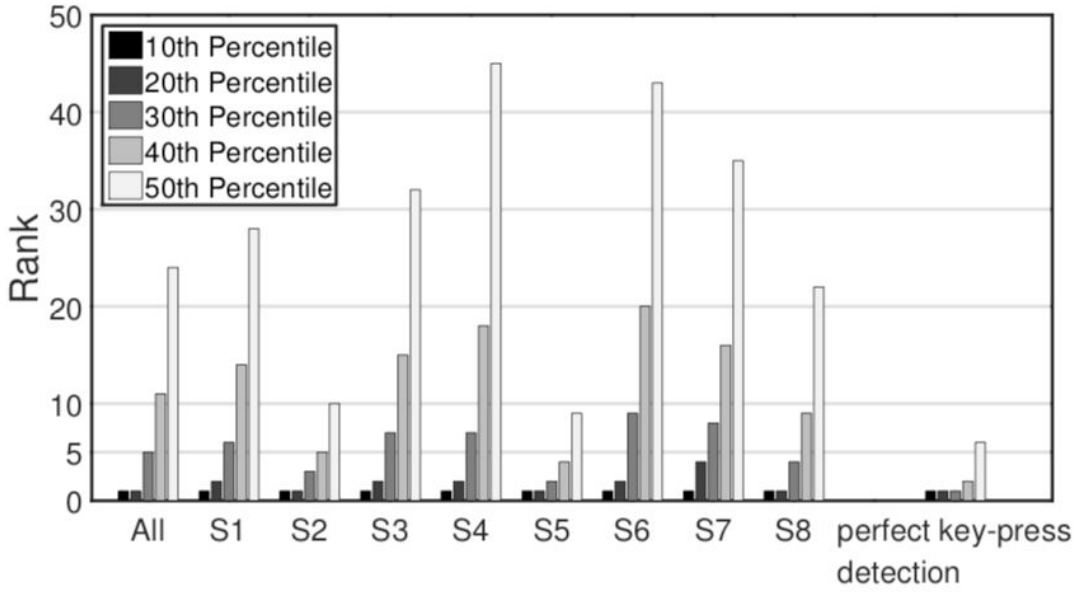


Figure 18: Rank of average, across users and with perfect key-press detection

With perfect keystroke detection data, rank drops sharply

Performance - What affects the rank

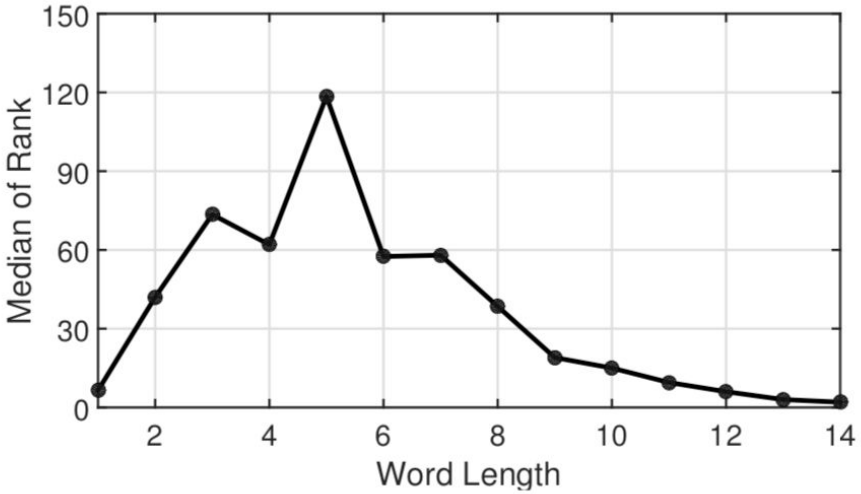


Figure 19: Median rank plotted against increasing word length – 4 – 7 length words show worse performance due to fewer keys to be detected while such words occur in large numbers.

Figure 20: Variation of rank against the number of characters typed by the left hand.

Performance - Impact of each Bayesian opportunity

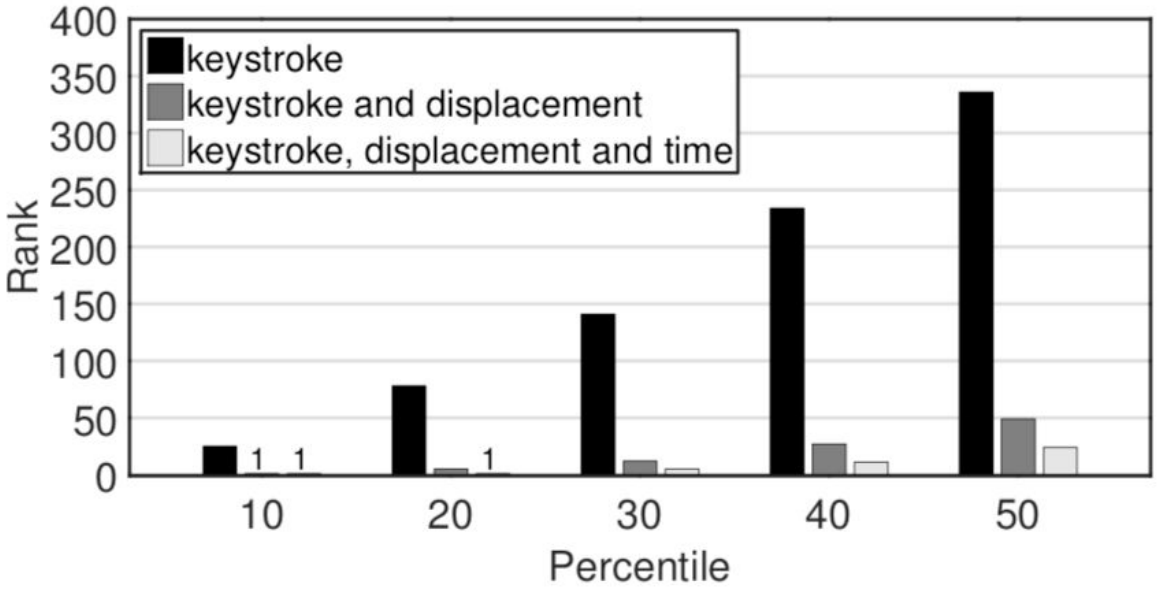


Figure 21: Contribution of different opportunities towards the overall performance of *MoLe*.

$$
\begin{aligned}
& P(O \mid W) \\
& = P(N \cap d_i, i = 1,2,...,N \cap t_j, j = 1,2,...,N-1 \mid W) \\
& \approx \sum_{(\alpha_1,...,\alpha_N)} P((c_{\alpha_1},...,c_{\alpha_N}) \mid W) \prod_{i=1}^{N} p(d_i \mid c_{\alpha_i}, c_{\alpha_{i-1}}) p((t_1,...,t_{N-1}) \\
& \mid (c_{\alpha_1},...,c_{\alpha_N}), (d_1,...,d_N), W)
\end{aligned}
$$

Performance - Impact of sampling rate

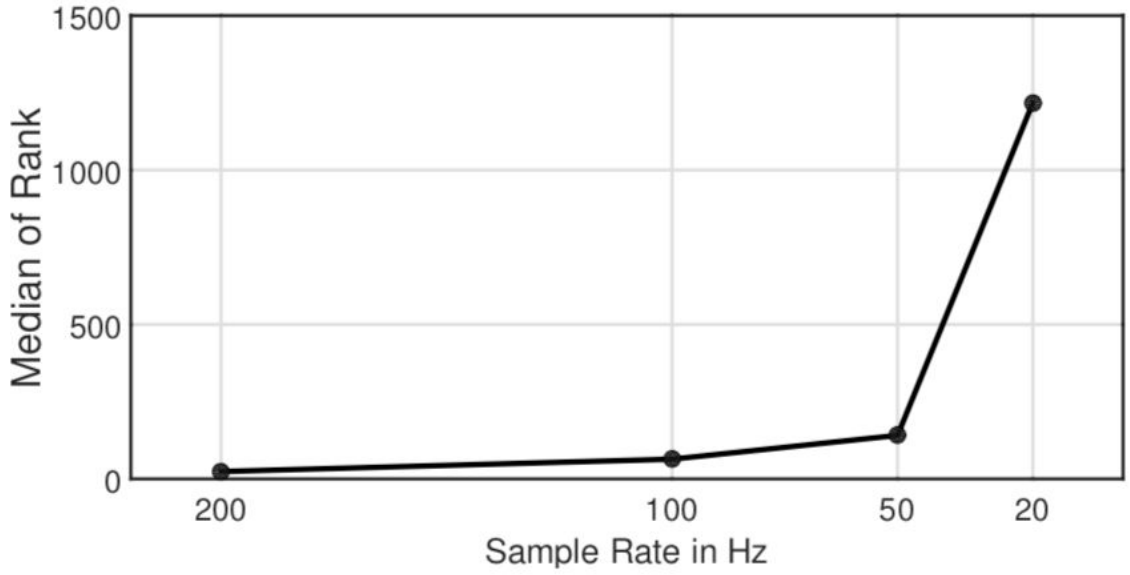


Figure 22: Lower sensor sampling rate rapidly reduces the ability to guess the word, perhaps indicating a way to thwart *MoLe*'s attack.

Conclusion - Contribution / Impact

- Identifying the possibility of leakage - required building blocks
  - key-press detection
  - hand-motion tracking
  - cross-user data matching
  - Bayesian inference
- Developing the system on Samsung Gear Live
  - experimenting with real users
  - reasonable accuracy
  - Sense of alarm

Conclusion - Limitation / future work

- Keyboard variant
- Confined to separate words
- inability to infer nonvalid English words, e.g. passwords
- Applying NLP/ human observation
- Typing activity classifier

Rank	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8
1.	motor	pistol	profound	technology	angel	those	that	disappear
2.	monitor	list	journalism	remaining	spray	today	tight	discourse
3.	them	but	originally	telephone	super	third	tightly	secondary
4.	the	lost	original	meanwhile	fire	through	thirty	adviser
5.	then	most	profile	headline	shore	towel	truth	discover

Table 1: ***Can you guess the correct sentence?*** **The words in each column are ranked in decreasing order of probability; also note that some words may not feature in the** $top-5$ **words presented in each column. The answer is made available at end of the paper[2].**