Exploring Privacy Risks of Inertial Sensing on Wearable Rings: A Feasibility Study on Keystroke Classification

Cody Arvonen Nastasia Maldei-Stumm Agniv Chatterjee
Aryan Jain Sujay Sanghavi Edison Thomaz
The University of Texas at Austin, Austin, TX, USA
codyarvonen@utexas.edu,n.maldeistumm@utexas.edu,agniv.chatterjee@utexas.edu
aryanj179@utexas.edu,sanghavi@mail.utexas.edu,ethomaz@utexas.edu

Abstract

The smart ring form factor is gaining popularity in wearable computing, as it offers a unique balance of stable, continuous signal acquisition and minimal obstruction to daily activities. Recent research has demonstrated that ring-worn Inertial Measurement Units (IMUs) are able to capture nuanced motor patterns, ranging from sign language gesture tracking and signature-based authentication to precise touch contact sensing on arbitrary surfaces. In this work, we investigated the feasibility of leveraging fine-grained finger movements to infer typed text on a keyboard using motion data from smart rings. This presents a potential privacy side-channel; if keystrokes can be inferred from a single ring alone, sensor access could be exploited to snoop sensitive user data such as passwords or private communications. To evaluate this threat, we implemented a pipeline leveraging a time-series foundation model to decode unconstrained typing. In a feasibility study with 2 participants, our method achieved an average top-1 keystroke accuracy of 84.5% from two wearable rings and successfully reconstructed semantically coherent English text without requiring strict typing constraints.

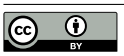
CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing**.

Keywords

Smart rings, Wearable IMUs, Keystroke prediction, Time-series foundation model, Side-channel privacy attack

ACM Reference Format:
Cody Arvonen Nastasia Maldei-Stumm Agniv Chatterjee, Aryan Jain Sujay Sanghavi Edison Thomaz . 2026. Exploring Privacy Risks of Inertial Sensing on Wearable Rings: A Feasibility Study on Keystroke Classification. In *The 4th International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems (HumanSys '26), October 26–30, 2026, Austin, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3842436.3843799



This work is licensed under a Creative Commons Attribution 4.0 International License.
HumanSys '26, Austin, TX, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2959-1/2026/10
https://doi.org/10.1145/3842436.3843799

1 Introduction and Motivation

While vision and audio sensing provide rich contextual data for ubiquitous computing, cameras and microphones in wearable applications pose well-documented privacy risks [6, 10]. In contrast, Inertial Measurement Units (IMUs) capture raw motion rather than explicit semantic content, making them widely trusted as a safer, privacy-preserving alternative for human activity recognition (HAR). However, as smartwatches have made wrist-worn IMUs ubiquitous, researchers have explored a security risk that is more subtle. Text entry on keyboards involves physical motion that could be picked up by these sensors, presenting an opportunity to snoop sensitive information like passwords or private messages.

Early investigations into this wearable side-channel include work by Sarkisyan et al. which leverages a smartwatch IMU to predict smartphone four-digit PINs and achieves greater than 41% top-5 accuracy [7]. A more challenging task is tackled by Maiti et al., who use wrist motion data to map inertial rotations and vibrations to keystroke transitions to shortlist typed words on a standard QWERTY keyboard, achieving greater than 50% top-10 accuracy for word classification [5]. While these results are surprising, they fail to demonstrate a compelling attack due to the compounding error inherent in sequence prediction. A low top-k accuracy means that enumerating every possible combination of characters or words quickly makes continuous text recovery practically infeasible. Even when supplementing wrist-worn IMUs with electromyography (EMG) to sense muscle activation, Gazzari et al. only achieve a 32% top-3 accuracy for classifying individual keystrokes on typed passwords [1]. Ultimately, while wrist and arm wearables validate the theoretical risk, they struggle to act as an accurate, standalone side-channel because the wrist moves significantly less than individual fingers during typing.

However, this side-channel may be better suited for a different form factor. Recently, smart rings have grown in popularity, with global annual shipments forecasted by the International Data Corporation to reach 1.7 million units in 2024 and 3.2 million by 2028 [9]. Smart rings have provided a powerful wearable form factor for HAR tasks due to the more fine-grained motion artifacts generated by the finger during activities involving the hands. For example, Liu et al. found

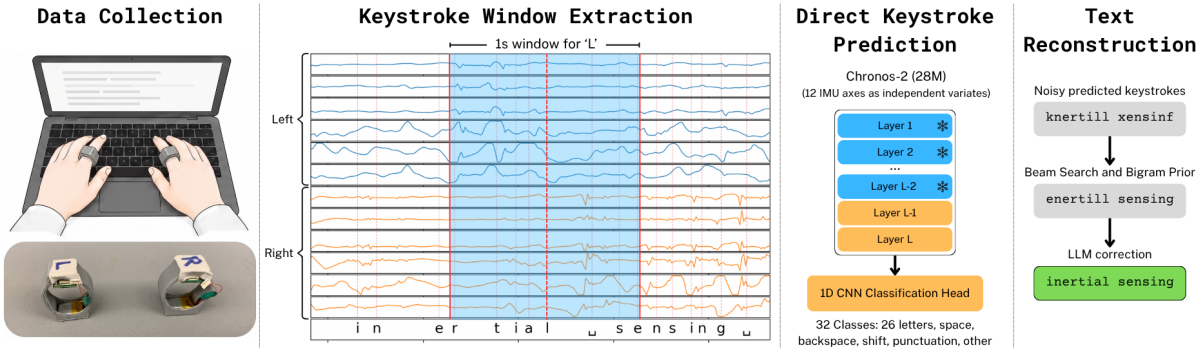


Figure 1: Pipeline for keystroke classification using two wearable ring IMUs and time-series foundation model feature extraction.

that an IMU on a ring wearable improved sign language gesture recognition compared to using only a smartwatch [4]. Since intricate finger dynamics are highly prevalent in typing, we believe that this privacy risk should be revisited in the context of smart rings. Previous works involving keystroke classification from wearable rings (typically one on each hand) do not consider this privacy concern but rather focus on a "virtual keyboard" application. As a result, studies that have been conducted involve data collected from users adhering to artificial typing speeds, following strict predetermined finger-to-key mappings, typing one character at a time, or using custom keyboard layouts [3, 12].

To address these limitations, we evaluated the feasibility of keystroke inference from smart rings as a privacy threat. Unlike prior work, our approach targets natural typing behavior on standard physical keyboards without enforcing constraints that may restrict natural typing motions. Although individuals typically wear only one smart ring, we targeted a system with two smart rings (one on each index finger) as a starting point to explore the correlation between finger motion dynamics and typed text. Our system is summarized in Figure 1, and our core contributions are as follows:

- **A Modern Approach to Keystroke Classification:** Instead of relying on traditional ML techniques for feature extraction as in previous works, we leveraged a time-series foundation model (Chronos-2) to extract meaningful temporal features from ring-based IMU data from a relatively small dataset.
- **LLM for Zero-Shot Text Reconstruction:** We implemented a training-free post-processing pipeline that utilizes a bigram language model and zero-shot Large Language Model (LLM) to reconstruct coherent English text from noisy keystroke probabilities.
- **Initial Empirical Validation:** We conducted a preliminary evaluation of our pipeline on a dataset of unconstrained typing from two participants wearing two ring devices, demonstrating an average top-1 keystroke accuracy of 84.5%.

2 Methodology

Data Collection. We collected a dataset of 123,165 keystrokes from two participants (authors of this work) wearing two smart rings each, one on each index finger. Each participant used a Python keylogging script on a MacBook keyboard to save the press and release timestamps of each keystroke. Two different types of sessions were recorded: *scripted sessions* where the participant typed out a pregenerated script and *naturalistic sessions* where the participant typed text impromptu. All sessions were recorded without imposing any restrictions on typing speed or finger-to-key mapping. Each script used was unique to that session and was not used in the scripted sessions of the other participant. P1 recorded six scripted sessions and seven naturalistic sessions, and P2 recorded seven scripted sessions. Three of the scripted sessions recorded by P1 were scripts generated by an LLM that was prompted to oversample words containing many rare characters ("k", "q", "j", "v", etc.) while still maintaining coherent and grammatically correct English. Across all sessions, P1 averaged 41.0 words per minute (WPM) with 83.1% efficiency[1], and P2 averaged 70.7 WPM with 95.5% efficiency.

Keystroke Classification. We trained a keystroke classification model directly on the Chronos-2 backbone, framing the task as a 32-class classification problem. These classes include the 26 English letters, the space key, backspace, shift, grouped punctuation, numbers, and an "other" category. Producing explicit character predictions ensured that the downstream language reconstruction stage relied entirely on the IMU-derived classifications rather than hallucinating text based on general language priors. We trained a CNN classification head on the outputs of the final transformer layer of `chronos-2-small` (28M parameters). The 12 axes (from the two 6-axis IMUs) were passed into Chronos-2 as independent variates, enabling cross-axis attention. To fully utilize the rich temporal feature space provided by Chronos-2, we employed a 1D CNN that traversed the time dimension rather

[1]Ratio of net final characters after deletions to total keystrokes

than relying on temporal global average pooling. Training samples were generated by extracting a fixed-size window (1s) of IMU data centered around the "press" timestamp of each keystroke. We froze the model weights of Chronos-2 except for the last two transformer layers that were fine-tuned during training. The model was trained using class-weighted cross-entropy loss for a maximum of 15 epochs, with an early stopping patience of 5 on the validation loss.

Text Reconstruction. During inference, the keystroke classification stage produces a noisy sequence of per-character probability distributions. We applied two sequential post-processing stages to recover coherent English text from this signal, neither of which involved any gradient-based training. First, we fit a character-level bigram language model on the ground-truth key token sequences from all training sessions. The model estimates the conditional probability $P(c_t \mid c_{t-1})$ using add-k smoothing ($k = 0.1$) over the 32-class character vocabulary. At inference time, we performed beam search with beam width $B = 5$, combining the per-keystroke log-probabilities from the keystroke predictor with the bigram log-probabilities via shallow fusion:

$$\text{score}(c_t) = (1 - \lambda) \log P_{\text{KP}}(c_t) + \lambda \log P_{\text{bigram}}(c_t \mid c_{t-1}) \quad (1)$$

where $\lambda = 0.3$ controls the relative contribution of the language model. At each timestep, the B active hypotheses were each expanded over the top-k candidates from the keystroke predictor, yielding up to $B \times k$ successors, which were then pruned back to the top B by cumulative score. Finally, the beam-decoded character sequence was passed to a large language model (GPT-5) with a zero-shot prompt instructing it to recover the most plausible intended text given the noisy input. No fine-tuning or in-context examples were used. Because the LLM may rephrase rather than transcribe verbatim, character error rate (CER) is not an appropriate primary metric for this stage. We instead report word error rate (WER) and BERTScore F1 [11], which capture word-level accuracy and semantic similarity, respectively.

Table 1: LOSO Results (Avg. across held-out sessions)

	Base LOSO		Personalized LOSO	
Participant	**Top-1 Acc.**	**Macro F1**	**Top-1 Acc.**	**Macro F1**
P1	0.7766	0.7326	**0.8472**	0.8074
P2	**0.9724**	0.9678	0.8277	0.7957
AVERAGE	**0.8451**	**0.8149**	**0.8404**	**0.8033**

3 Results

Cross Validation. We performed a leave-one-session-out (LOSO) cross-validation to analyze the performance of our model on each individual session. We performed the following two LOSO studies:

- **Base LOSO:** We held out one session as the test set and trained on all other sessions from both participants.
- **Personalized LOSO:** We held out one session as the test set and trained only on the other sessions belonging to the participant of the held out session (i.e., each participant was trained using only their own data).

For both LOSO types, a validation set for early stopping was taken from the train set using a random 80/20 split. Table 1 displays the results of both cross-validation runs. For the base LOSO our model achieved a top-1 accuracy of 0.8451 and a macro F1 score of 0.8149 as averages across all held out sessions. We note that P2 outperformed P1 on the base LOSO evaluation, but not on the personalized LOSO evaluation. We hypothesize that in the personalized LOSO, P2 performed worse because it no longer had access to the scripted sessions from P1 with rare keys in its training data, whereas P1 performed better because it was able to learn only from its own data.

Table 2: Performance metrics (Acc., WER, and BERTScore)

	Keystroke Pred.	*Beam+Bigram*	*LLM*	
Session	**Acc.**	**WER**	**WER**	**BERTScore**
7 (P2)	98%	0.142	0.085	0.272	0.991
11 (P1)	83%	0.690	0.592	0.391	0.901
8 (P1)	62%	0.911	0.853	0.730	0.848

Semantic Recovery. We evaluated the text reconstruction pipeline on three representative sessions from the base LOSO cross-validation, selected to span the range of keystroke prediction accuracy: high (98% top-1), medium (83% top-1), and low (62% top-1) accuracy. Results are shown in Table 2. Beam search with bigram language model fusion consistently reduced WER over the raw keystroke prediction output across all three sessions. LLM zero-shot correction further reduced WER in the medium and low accuracy sessions. BERTScore F1 remained high across all sessions (0.848–0.991), indicating that semantic content was well-preserved even when surface-level character accuracy was low.

Theoretical Upper Bound on Performance. To further explore the feasibility of a legitimate privacy leakage, we tested the theoretical limit of reconstructing text from only a single ring. We took the text from the ground truth typed data and masked out the characters corresponding to either the left-hand side ("q, w, e, r, t, b, a, s, d, f, g, z, x, c, v") or the right-hand side ("y, u, i, o, p, h, j, k, l, n, m") of the keyboard with a single blank character ("_") denoting each missing letter. The left-masked and right-masked data from each session was then passed into an LLM (`gemini-3.1-flash-lite`), prompting it to find the most probable actual text. The space bar and punctuation were not masked for this test, and we excluded the sessions from P1 that were oversampled for rare characters. We report an average WER and BERTScore F1 of 0.089 and 0.975 for right-masked data (ring worn on left hand) and 0.284 and 0.926 for left-masked data, respectively.

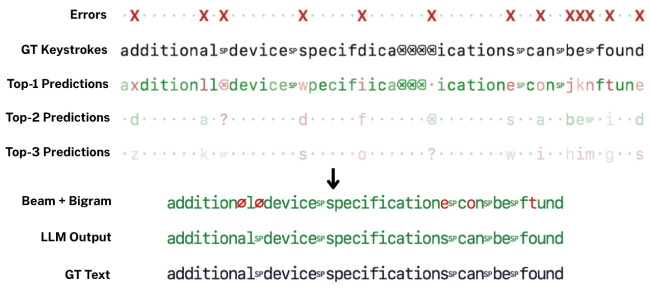


Figure 2: Example output for a snippet of session 11 from P1.

This indicates that LLMs can accurately reconstruct text from the characters on only one side of the keyboard. This assumes that an accurate count of missing letters is kept, which has been shown to be feasible by leveraging acoustic signals for keystroke presence detection [2].

4 Discussion and Conclusions

Overall, our approach performed surprisingly well on a challenging task. Figure 2 shows an example output from P1 session 11 at each stage of the keystroke prediction to text reconstruction pipeline. With 75.5% top-1 accuracy in keystroke prediction for this particular snippet, we were able to reconstruct the entire text sequence.

Common Misclassifications. The confusion matrix in Figure 3 shows that the vast majority of misclassifications occur on character pairs like o/i, e/r, s/a, t/g, and i/u that are adjacent to each other on the keyboard. Additionally, we note that for 2 of the 13 misclassifications in Figure 2, the prior or subsequent character was predicted instead of the current one. For example, in the first word "additional", our model predicted "ll" instead of "al" with relatively high confidence. We attribute this to how similar sample windows are for two keys typed in quick succession. When typing, two keys are often pressed only milliseconds apart, resulting in two nearly identical windows that are slightly shifted versions of each other but represent different characters at different locations on the keyboard. Despite these common mistakes, we note that many of the misclassifications were predictions with lower confidence produced by the model, and often the correct character was only the second or third guess.

Limitations and Future Work. While our method achieved high performance, we acknowledge that this work is still preliminary. With only two participants, it is difficult to evaluate generalizability to unseen participants due to the unique characteristics of individual typing dynamics [8]. Additionally, while our initial investigations focused on keystroke classification from two ring wearables instead of one, we presented the first ring-based system capable of reconstructing typed text in naturalistic typing environments. Further investigations involving more participants and single-ring

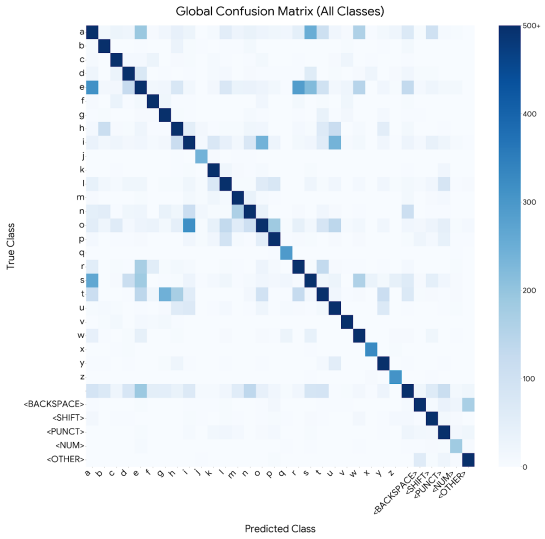


Figure 3: Confusion matrix for keystroke classification.

evaluation would strengthen the results of this work and the feasibility of this privacy concern.

Acknowledgments

This research was funded by the NIDDK and NIH, grant number 5R01DK134546-02.

References

[1] M. Gazzari, A. Mattmann, M. Maass, and M. Hollick. My(o) Armband Leaks Passwords: An EMG and IMU Based Keylogging Side-Channel Attack. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(4):157:1–157:24, Dec. 2022.

[2] G. Li, Y. Arakawa, Y. Nakamura, H. Choi, S. Fukushima, and W. Wang. WatchLogger: Keyboard Typing Words Recognition Based on Smartwatch. In *2023 Fourteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pages 1–6, Nov. 2023.

[3] C. Lian, X. Ren, Y. Zhao, X. Zhang, R. Chen, S. Wang, X. Sha, and W. J. Li. Towards a Virtual Keyboard Scheme Based on Wearing One Motion Sensor Ring on Each Hand. *IEEE Sensors Journal*, 21(3):3379–3387, Feb. 2021.

[4] Y. Liu, F. Jiang, and M. Gowda. Finger Gesture Tracking for Interactive Applications: A Pilot Study with Sign Languages. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(3):112:1–112:21, Sept. 2020.

[5] A. Maiti, O. Armbruster, M. Jadliwala, and J. He. Smartwatch-Based Keystroke Inference Attacks and Context-Aware Protection Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 795–806, New York, NY, USA, May 2016. Association for Computing Machinery.

[6] D. Major, D. Y. Huang, M. Chetty, and N. Feamster. Alexa, who am i speaking to?: Understanding users' ability to identify third-party apps on amazon alexa. *ACM Trans. Internet Technol.*, 22(1), Sept. 2021.

[7] A. Sarkisyan, R. Debbiny, and A. Nahapetian. WristSnoop: Smartphone PINs prediction using smartwatch motion sensors. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Nov. 2015.

[8] D. Senarath, S. Tharinda, M. Vishvajith, S. Rasnayaka, S. Wickramanayake, and D. Meedeniya. BehaveFormer: A Framework with Spatio-Temporal Dual Attention Transformers for IMU-enhanced Keystroke Dynamics. In *2023 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9, Sept. 2023. ISSN: 2474-9699.

[9] F. Stanbrell. The future of smart rings. https://www.idc.com/resource-center/blog/the-future-of-smart-rings/, October 2024. Accessed: June 23, 2026.

[10] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia. Placeavoider: Steering first-person cameras away from sensitive spaces. 01 2014.

[11] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.

[12] Y. Zhao, X. Ren, C. Lian, R. Ma, X. Zhang, X. Sha, and W. J. Li. A Smart Wireless IoT Ring for Real-Time Keystroke Recognition Using Edge Computing. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13, 2022.