

A Comparative Study of Regression Techniques for Touch-based Input and Adaptive User Interfaces

Irina PETRARIU¹, Laura-Bianca BILIUS², Liviu MITROFAN¹, Sebastian CRINTEA¹, Robert CROITOR¹, Tudor HOROMNEA¹

¹ASSIST Software, ²MintViz Lab | MANSiD Research Center, Stefan cel Mare University of Suceava
Suceava, Romania

irina.petrariu@assist.ro, laura.bilius@usm.ro, liviu.mitrofan@assist.ro,
sebastian.crintea@assist.ro, robert.croitor@assist.ro, tudor.horomnea@assist.ro

Abstract—Touch interactions are integral to modern devices. Also, emerging AI-generated interfaces offer new opportunities for personalized user experiences. In this context, we conducted an empirical study with 23 participants to understand which touch-related features are needed to facilitate generation of adaptive user interfaces. We extracted the two most relevant features (normalized distance and whether the touch was inside the button) and compared five regression algorithms. We found that Ridge Regression (RR) and Passive Aggressive (PA) yielded the best results under leave-one-out cross-validation testing, and RR exhibited the most consistent performance under time series split. We also evaluated the RR model with two additional participants with upper-body motor impairments and found that model performance depends on individual motor abilities.

Keywords—touch input; regression models; feature selection; wearables

I. INTRODUCTION

The widespread adoption of touchscreen smart devices [1], [2], ranging from smartphones and smartwatches to tablets and laptop screens, has facilitated the acceptance of gesture-based interaction techniques, including tapping, dragging, swiping, and pinching [3]. Among these, tapping remains the most frequently used gesture [3]. Consequently, the research community has investigated touch-based interaction techniques with the goal of improving the overall user experience [4], [5], [6], [7] and understanding the impact on various user categories [8], including users with motor impairments [9], [10].

Guidelines and industry standards have been established to support the design of effective touchscreen user interfaces. For example, the minimum recommended button size is 10mm [11]. However, prior work highlighted that context could change how users interact with interfaces. For instance, walking triggers a very high number of errors, especially in smaller buttons [12]. Moreover, recent advances in AI techniques have enabled the possibility of automatically generating user interfaces [13], [14], [15] based on specific user needs, preferences, and context. Nonetheless, such personalization requires access to user-specific information to enable adaptive behavior. Interaction performance is influenced by individual differences such as motor control [9], age [8], and situational impairments [12], which are not captured by generic design guidelines. This variability highlights the need for adaptive interfaces that can dynamically adjust parameters such as target size, spacing, or feedback. Personalization is beneficial for both usability and for ensuring accessibility across diverse users.

In this regard, we conducted an empirical study with 23 participants in 5 sessions. During our study, we extracted 5 different features to describe touch input, using an application developed in Unity that consists of a button that moves randomly following each tap. We conducted an analysis of the selected 5 features using a 1NN regressor with a 5-fold cross-validation to find the most relevant features, normalized distance and if the touch happened inside the button. Using this outcome, we continued with an evaluation of 5 regression models using Leave-One-Out Cross-Validation (LOOCV) and Time Series Split (TSS). Given that our results showed low errors, we analyzed the impact on participants with specific motor abilities.

In this paper, Section II presents prior work related to touch-based interaction techniques. Section III provides a description of the database, the feature selection process, and the applied evaluation techniques. Section IV includes the results of the feature selection and the results obtained by evaluating the regression models. We continued this section by further discussing our results, using data collected from two additional participants with upper-body motor impairments, drawing out a conclusion in Section V.

II. RELATED WORK

Given that tapping is the most frequently used touch-based interaction technique [3], the research community has investigated a range of related topics, starting with increasing touch recognition accuracy [4], [16], reducing latency [5] and error rates [7], and up to analyzing target dimensions [6]. For example, Usuba et al. [4] proposed a model that predicts how accurately a user would successfully tap a 2D rectangular UI element using data collected from 1D pointing tasks (touching a vertical or horizontal bar), which resulted in a slightly lower success rate than training with 2D data. Their approach built on the Dual Gaussian distribution of touch points, assuming independent variability along the horizontal and vertical axes, and decomposed success rate into the product of two 1D probabilities. Nishida et al. [5] introduced PredicTaps, a system that employed machine learning to reduce the time to detect a single tap from a double tap from 150–500 ms to 12–17.6 ms. More specifically, PredicTaps addressed the inherent single-tap latency problem caused by conventional gesture recognition techniques, where systems must wait for a predefined double-tap threshold before confirming a single tap. Huang et al. [7] developed SpecIFingers, a system that differentiates between the hand fingers to enhance finger-specific interactions, with a mean accuracy over 85%. They further improved overall interaction reliability by introducing an explicit correction mechanism triggered by placing all five

fingers on the screen, enabling a complete pipeline accuracy of 99.45%.

Regarding target design, Raidah et al. [6] investigated errors in touchscreen interactions and target dimensions to understand why users make mistakes and identify patterns. Their findings show that the finger-to-target size ratio is key, and increasing the size of the objects reduces error probabilities. In particular, their work emphasized that touch input cannot be treated as a single-point interaction, since the fingertip has a physical area that varies across users, which directly affects accuracy. Additionally, Zheng et al. [8] proposed a model that achieves high accuracy in predicting error rates when touching moving targets, and highlighted the impact of low motor ability, which affects the elderly and children. Their work introduces a scenario-adaptive modelling approach that extends the traditional Ternary-Gaussian framework by allowing it to adapt to specific user conditions and interaction contexts using only a small amount of new data (as few as 20 endpoints).

Moreover, employing Generative AI (GenAI) to create user interfaces is an emerging domain where guidelines are still to be established. For instance, Lee [13] highlights that this new technology is creating a paradigm shift in how interfaces are designed. They proposed a working definition of a Generative User Interface as a collaboration between human input and Artificial Intelligence (AI) and discussed multiple challenges, including value prioritization and balance, bias and representation, and attribution and intellectual property, to name a few. Sun [14] showcased that GenAI can be adaptive towards the individual designers' styles and preferences. They ran an experimental evaluation that compared AI-assisted design to manual design. Their results showed that AI-assisted design was 26% faster than manual design and reduced the overall mental load. In addition, the prototypes designed with AI were perceived as more creative and with a slightly more diverse layout. Also, as emphasized by Deriba et al. [15], GenAI can also improve accessibility awareness by warning designers of issues, such as poor contrast, inaccessible forms, etc., and suggesting improvements, such as keyboard alternatives and using audio recognition. They identified that tools limitations and accessibility knowledge are a primary challenge in GenAI efficacy, because the response is highly dependent on the precision and specificity of the posed questions.

While generated user interfaces are generally efficient, users with specific abilities, such as users with upper-body impairments, represent a distinct target group that requires adaptive design. For instance, Findlater et al. [9] showed that users with motor impairments exhibit three times larger error rates with touch input compared to mouse input. Their controlled study comparing touchscreen and mouse interactions across multiple tasks (e.g., pointing, dragging, and steering) further revealed that, although touch input can be faster overall, it does not provide the same accuracy benefits for users with motor impairments as it does for users without impairments. Vatavu and Ungurean [17] reported that stroke gestures performed on touchscreens by users with motor impairments are recognized less accurately than those produced by users without impairments, but can nevertheless be robustly synthesized with kinematic profiles [18], which opens opportunities for generative techniques in user modeling and training recognizers. Moreover, Sean and Vatavu [10] highlighted the lack of involvement of users with

motor impairments in user studies about wearable interaction. Their systematic review of 57 papers revealed that nearly half of the works did not include participants with motor impairments, and when they were included, studies typically involved only a limited number of participants (median of six), raising concerns about the representativeness and generalizability of findings.

In this work, we build on this prior work leading to generative AI-driven user interface adaptation, which we complement with new empirical findings involving various touch input features and recognition models. In the following, we will present our methodology.

III. METHODOLOGY

A. Description of the database

Twenty-three (23) volunteers (18 male, 5 female) participated in the data collection process, ages between 21 and 42 ($M=27.86$, $SD=5.24$). All participants used smartphones on a daily basis, with self-reported screen times ranging from 1 hour to 5 hours ($M=2.6$, $SD=1.42$). The sample size of 23 participants aligns with Human-Computer Interaction (HCI) research norms, exceeding the median typically reported in studies, where the most common sample size is 12 [19], [20].

To support participant engagement, we developed a custom Android mobile application, developed in Unity, that integrates gamification elements, as seen in prior work [16]. The participants had to touch a button displayed at random locations on the screen in five different sessions. The button size was approximately 1.5cm, relative to the screen resolution, which varied from 1080×1920 to 1440×3216 . Participants were instructed to mimic normal gaming interactions with breaks between sessions. Each session consisted of $N=50$ touch points, which resulted in 250 touch interactions recorded for each participant. Thus, we recorded a total of $23 \times 250 = 5750$ touch interactions used in the analysis. Unlike prior empirical analyses [21], which focused on understanding training size requirements for modeling user input with touch time and offset measurements, we considered a wider array of touch-related features, as follows:

C_1 DURATION, in milliseconds, as the interval between the first finger-down and finger-up events;

C_2 NORMALIZED_DISTANCE, as the Euclidean distance between the touch location and the center of the button, normalized to the button size;

C_3 IS_INSIDE_BUTTON, binary value, if the touch was inside the button (value 1) or outside the button (value 0).

C_4 RELATIVE_X_POSITION_IN_BUTTON, values between 0.0 and 1.0, which denotes the touch location's horizontal position in the button.

C_5 DELTA_X_FROM_CENTER, in pixels, as the touch location's horizontal deviation relative to the button center.

B. Features Selection

To determine which features were the most relevant, we evaluated the performance of all possible combinations of the extracted features. Specifically, we generated every subset of the extracted features and assessed their performance using a 1-Nearest Neighbor (1NN) regressor with the Euclidean Distance. The 1NN was chosen because it is commonly used

as a baselined method, serving as a reference point for evaluating algorithms in machine learning [22], [23]. For this evaluation, a k-fold cross-validation with five folds (5-fold CV) was performed.

C. Evaluation Techniques

After we selected the most relevant features, we computed several regression algorithms from the Scikit Learn package [24]. We selected the following regression algorithms:

A₁ Nearest Neighbor (NN) predicts target values by finding a defined number of training samples closest in proximity to a new point [22].

A₂ Decision Trees (DT) builds a flowchart-like model by recursively splitting training data into subsets based on feature values to maximize information gain or reduce variance [25].

A₃ Ridge (RR) is a linear regression variant that introduces an L₂ penalty to the size of the coefficients to reduce the effects of highly correlated predictor variables [26].

A₄ Support Vector Machine (SVR) maps data to a high-dimensional feature space using kernels to perform linear regression while maintaining errors within a specific threshold margin [27].

A₅ Passive Aggressive (PA) operates in an online setting where it makes no model updates if a prediction is accurate (passive) but rapidly adjusts parameters to resolve a loss if the threshold is violated (aggressive) [28].

The regression models selected for this study cover different modeling perspectives including linear relationships (RR), hierarchical decision structures (DT), spatial separation (SVR), instance-based learning (NN), and incremental learning (PA) [29], [30], [31]. All algorithms were executed using the default parameters provided by the Scikit-learn library [24] focusing on the differences between the algorithms themselves [32]. Furthermore, Weerts et al. [33] mentioned that default configurations in Scikit-learn often yield non-inferior performance compared to tuned models across diverse datasets. To evaluate the regression models on our data, we used two validation techniques, as follows:

- a) Leave-One-Out Cross-Validation (LOOCV): evaluates the model by iteratively leaving one sample out for testing while training on the remaining data [34]. The LOOCV evaluation was performed independently for each participant, using data from all sessions together.
- b) Time Series Split (TSS): a cross-validation strategy adapted for temporal data. TSS preserves a chronological order of the data by training the model on past observations and testing it on new samples [35]. The TSS evaluation was performed independently for each participant. Specifically, each participant's sessions were ordered chronologically, and the model was trained on earlier sessions while being evaluated on subsequent sessions from the same participant.

Model performance was assessed using the following measures:

- a) Root Mean Squared Error (RMSE) [36]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (1)$$

- b) Mean Absolute Error (MAE) [36]:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i| \quad (2)$$

- c) Absolute Error (AE):

$$AE = |\bar{y}_i - y| \quad (3)$$

Where $y_i, i = 1, n$ are the observations and $\bar{y}_i, i = 1, n$ represent the model predictions.

IV. RESULTS

A. Feature Selection

To identify the most relevant features, we evaluated the combinations of the five extracted features from our participants. Specifically, we computed a total of $2^5-1=31$ combinations (e.g., {DURATION}, {DURATION, NORMALIZED DISTANCE}, {DURATION, NORMALIZED DISTANCE, RELATIVE X POSITION IN BUTTON}), and for each combination, a 5-fold CV was performed. Table I presents the top 10 feature combinations obtained during the feature selection process. The combinations are ranked according to their predictive performance measured using RMS and MAE.

TABLE I. TOP 10 COMBINATIONS OBTAINED DURING FEATURE SELECTION USING 5-FOLD CV (RMSE AND MAE ARE REPORTED).

Features	RMSE (avg ± std)	MAE (avg ± std)
[C ₂ , C ₃]	0.0129 ± 0.0059	0.0010 ± 0.0003
[C ₁ , C ₂]	0.0366 ± 0.0103	0.0039 ± 0.0004
[C ₁ , C ₂ , C ₃]	0.0369 ± 0.0101	0.0042 ± 0.0005
[C ₁ , C ₂ , C ₅]	0.0522 ± 0.0123	0.0100 ± 0.0014
[C ₁ , C ₂ , C ₅ , C ₃]	0.0522 ± 0.0124	0.0100 ± 0.0015
[C ₁ , C ₂ , C ₄ , C ₃]	0.0523 ± 0.0124	0.0099 ± 0.0015
[C ₁ , C ₂ , C ₄]	0.0523 ± 0.0124	0.0099 ± 0.0015
[C ₂ , C ₄ , C ₃]	0.0527 ± 0.0076	0.0058 ± 0.0011
[C ₂ , C ₄]	0.0527 ± 0.0075	0.0057 ± 0.0011
[C ₂ , C ₅]	0.0528 ± 0.0076	0.0058 ± 0.0011

As depicted in Table I, among the evaluated feature combinations, the best performance (RMSE = 0.0129, MAE = 0.0010) was obtained using [C₂, C₃], specifically NORMALIZED_DISTANCE and IS_INSIDE_BUTTON features. NORMALIZED_DISTANCE reflects the spatial accuracy of touch interaction, while IS_INSIDE_BUTTON indicates whether the interaction occurred within the target area, concluding that these two features provide sufficient information to accurately predict our data. Moreover, this result indicates that distance and target inclusion capture the main characteristics of touch interaction. Additional features do not improve performance; therefore, the prediction task can be simplified without loss of accuracy. These features also reduce model complexity and computational cost.

B. Evaluation of Regression Models

The overall LOOCV RMSE values obtained across all regression algorithms indicate differences in the predictive performance of the evaluated regression models. According to Table II, our results indicate that RR (RMSE=0.0494) and PA (RMSE=0.0498) achieved the best overall generalization performance under LOOCV technique. Also, it can be noted that the results are not specific to a single user, and this pattern can be observed across all participants; see Fig. 1.

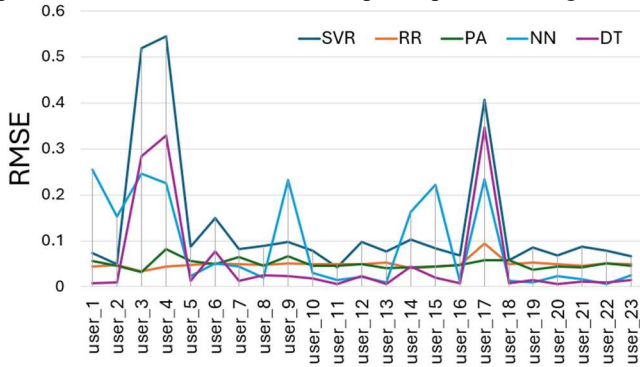


Fig. 1. RMSE values obtained for each participant using Leave-One-Out Cross-Validation across regression algorithms evaluated in this paper.

In Fig. 2, the RMSE values are depicted for the evaluated regression algorithms across the five sessions using TSS. Each session included the previous ones, meaning that the dataset for every participant includes all earlier observations along with the current data, as follows: Split 1 includes data collected from Session 1 used for training and data collected from Session 2 used for testing, Split 2 includes Sessions S1 and S2 for training and Session 3 for testing, Split 3 includes Sessions S1, S2 and S3 for training and Session 4 for testing, and Split 4 includes Sessions S1, S2, S3 and S4 for training and Session 5 for testing.

TABLE II. LOOCV AND TSS OVERALL RMSE AND MAE VALUES ACROSS REGRESSION ALGORITHMS EVALUATED IN THIS PAPER.

		SVR	RR	PA	NN	DT
LOOCV	RMSE	0.1345	0.0494	0.0498	0.0890	0.0571
	MAE	0.0737	0.0346	0.0395	0.0122	0.0112
TSS	RMSE	0.0718	0.0006	0.0319	0.0067	0.0039
	MAE	0.0530	0.0004	0.0284	0.0014	0.0009

The performance of the evaluated regression algorithms, assessed using both LOOCV and TSS, reveals some differences; see Table II. RR achieves the lowest errors across both LOOCV and TSS techniques, RMSE=0.0494 and MAE=0.0346, respectively, RMSE=0.0006 and MAE=0.0004. NN and DT show low MAE values even when RMSE is moderately higher in LOOCV, highlighting their ability to make precise predictions.

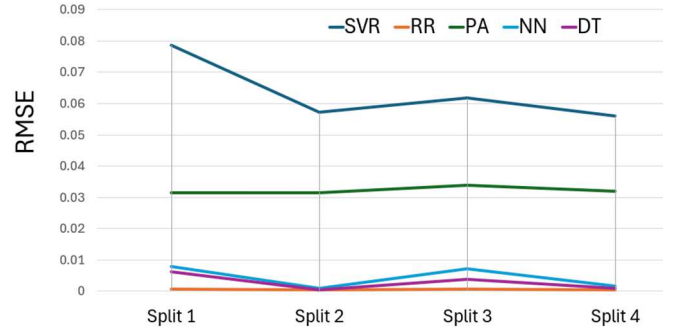


Fig. 2. Average RMSE values across all participants for regression algorithms evaluated in this paper over five sessions using TSS, where each split incrementally includes previous sessions for training and evaluates performance on the subsequent session.

TSS results illustrate that NN and DT improve across sessions, suggesting these algorithms can leverage cumulative data to refine temporal predictions (RMSE: NN=0.0067, DT=0.0039; MAE: NN= 0.0014, DT=0.0009). In contrast, RR consistently achieved the lowest errors while remaining stable across the five sessions; see Fig. 2. Overall, the performance of models under temporal evaluation (TSS) indicates that user interaction patterns can be learned progressively. This opens opportunities for incremental and online adaptation strategies, where systems refine their behavior as more interaction data becomes available. Such approaches are particularly relevant for wearable and mobile devices, where interaction contexts frequently change and personalization can significantly enhance user experience.

Considering both LOOCV and TSS results, RR generalizes across both participants and through time. These results are reinforced by the feature selection which indicated that `NORMALIZED_DISTANCE` and `IS_INSIDE_BUTTON` are sufficient to achieve a high predictive performance.

C. Evaluation with users with motor impairments

Given the widespread integration of touchscreens in a great diversity of smart devices, such as smartphones, smartwatches, or laptops, user interfaces should be adaptable to specific motor abilities [9], [21]. In this context, we evaluated the RR algorithm using data collected from two participants with upper-body motor impairments. Participant P1 (48 years) reported having traumatic brain injury and difficulties with coordination, whereas participant P2 (61 years), with spinal cord injury at the T7 level, did not report poor coordination. The participants recorded 55 taps during one session in our application. This data was predicted using the RR model trained on the data collected from the 23 participants without upper-body motor impairments. This represents a mixed training approach designed to evaluate how well the model predicts and generalizes data collected from participants with specific abilities. In Fig. 3, we report the AE between the observed and predicted values.

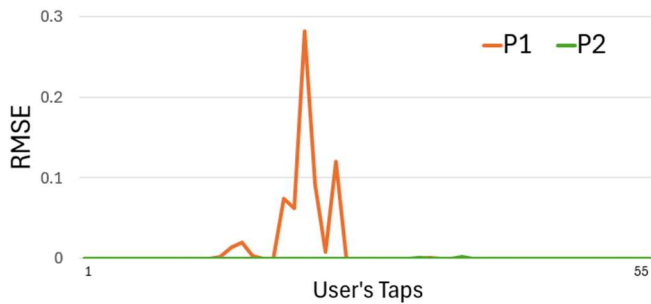


Fig. 3. Absolute error values for P1 and P2 participants across 55 taps collected in a session.

The analysis of AEs shows differences in how the model performs for the two participants; see Fig. 3. For participant P2, who did not report coordination difficulties, the errors are smaller, which is also reflected in a very low RMSE value (0.0003). In contrast, participant P1 shows greater variability, with larger absolute errors due to less consistent touch behavior associated with coordination difficulties leading to a higher RMSE (0.0451).

The variability observed for participant P1 highlights how health condition and specific motor impairments can affect prediction accuracy. In contrast, the lower error observed for participant P2 indicates that the level of motor variability, rather than the presence of an impairment, influences model performance. This suggests that models trained exclusively on data from users without motor impairments may not generalize well to a more diverse way in which users with motor impairments interact with touch interfaces. Overall, while the RR model demonstrates an ability to generalize across mixed training data, its performance is influenced by health condition and specific motor impairments.

D. Discussion

The findings of this study have implications for both designers and developers of touch-based systems. The observed differences in interaction patterns indicate that users do not show uniform touch behavior, particularly in the presence of motor impairments. As a result, systems designed based on assumptions of consistent and precise input may not perform equally well for all users. Therefore, interfaces should be more tolerant to input variability. This can be achieved by increasing target sizes, improving spacing between interactive elements, and reducing the dependence on precise gestures. Also, predictive models and interaction algorithms should incorporate flexibility to handle inconsistent input patterns. Furthermore, the results highlight the importance of adaptive systems that can adjust to individual user behavior over time. Incorporating personalization mechanisms may improve usability for users with motor impairments, and not only [37].

E. Limitations and Future Work

This study has several limitations that inform directions for future research. First, our application used a single button size of approximately 1.5 cm, which may limit the generalizability of the findings. Future work should investigate whether various button sizes and layouts influence the relevance of the identified features, specifically

in relation to touch accuracy. Second, the evaluation involving participants with upper-body motor impairments was limited to only two users. While this was sufficient for an initial assessment of model behavior, it does not support generalizations about this population. Future studies should include a larger and more diverse group of users with varying types and levels of motor impairments, as well as longitudinal data collection to better capture intra-user variability over time. Additionally, the data collection process was conducted in a static and controlled environment. However, prior work indicates that contextual factors, such as mobility (e.g., walking), device usage conditions, or environmental distractions, can significantly influence touch interaction performance [12]. Therefore, future work should evaluate the proposed models in real-world scenarios. Another limitation is related to the diversity of devices used during data collection. Although screen resolutions varied across participants, other device-specific factors such as touch sensitivity, sampling rate, and hardware differences were not explicitly controlled or analyzed. These aspects may introduce variability in the recorded data and should be investigated in future research. Finally, all regression models were evaluated using default parameter settings provided by the Scikit-learn library [33]. While this approach allows for a fair baseline comparison, it may not reflect the optimal performance of each model. Future work should explore hyperparameter tuning and model optimization techniques to further improve predictive accuracy and better understand the trade-offs between model complexity and generalization.

V. CONCLUSION

In this study, we explored regression models to predict touch interaction behavior using data collected from 23 participants. Feature selection showed that normalized distance and whether the touch was inside the target area were sufficient for high predictive performance. Using these two features, we evaluated five different regression algorithms to assess their performance on our data. Ridge Regression and Passive Aggressive performed best under LOOCV, but RR yielded the most consistent performance across sessions when evaluated with TSS. Testing the best model (RR) on additional participants with upper-body motor impairments revealed that performance depends on individual motor abilities, highlighting the importance of an ability-based approach that adapts to the user rather than requiring the user to adapt to a standard interface.

ACKNOWLEDGMENT

This work was supported by a grant of the Ministry of Education and Research, CCCDI-UEFISCDI, project number PN-IV-P7-7.1-PTE-2024-0434, within PNCDI IV. The authors would like to thank Ovidiu-Ciprian Ungurean for his assistance with data collection from participants with motor impairments.

REFERENCES

- [1] Statista, "Number of smartphones sold to end users worldwide from 2007 to 2023," *statista.com*, Jan. 2023. [Online]. Available: <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/> [Accessed: Mar. 2026].

- [2] Technavio, “Touch screen market analysis APAC, North America, Europe, South America, Middle East and Africa - US, China, Japan, India, Germany - Size and forecast 2024–2028,” *technavio.com*, Apr. 2024. [Online]. Available: <https://www.technavio.com/report/touch-screen-market-industry-analysis> [Accessed Mar. 2026].
- [3] J. Breuninger, “Suitability of touch gestures and virtual physics in touchscreen user interfaces for critical tasks,” Ph.D. Thesis, Technische Universität München, Munich, Germany, 2020, doi: [10.13140/RG.2.2.16704.15363](https://doi.org/10.13140/RG.2.2.16704.15363).
- [4] H. Usuba, S. Yamanaka, J. Sato, and H. Miyashita, “Predicting touch accuracy for rectangular targets by using one-dimensional task results,” in *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. ISS, Art. no. 579, pp. 525–537, Dec. 2022, doi: [10.1145/3567732](https://doi.org/10.1145/3567732).
- [5] N. Nishida, K. Ikematsu, J. Sato, S. Yamanaka, and K. Tsubouchi, “Single-tap latency reduction with single- or double-tap prediction,” in *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. MCHI, Art. no. 224, pp. 1–26, 2023, doi: [10.1145/3604271](https://doi.org/10.1145/3604271).
- [6] F. N. Raidah, K. F. N. Sohada, K. T. Hasan, and M. M. T. Rahman, “Optimization of touch interface design by analyzing error probability and their correlation with fingertip and object dimensions: Fitts’ law: Touch screen design optimization,” in *Proc. 3rd Int. Conf. Comput. Advancements (ICCA ’24)*, New York, NY, USA: ACM, 2025, pp. 247–256, doi: [10.1145/3723178.3723211](https://doi.org/10.1145/3723178.3723211).
- [7] Z. Huang *et al.*, “SpeciFingers: Finger identification and error correction on capacitive touchscreens,” in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 1, Art. no. 8, 2024, pp. 1–28 doi: [10.1145/3643559](https://doi.org/10.1145/3643559).
- [8] Y. Zheng, J. Huang, J. Liu, C. Yang, and F. Tian, “A scenario adaptive model for predicting error rates in moving target selection on smartphones,” in *Proc. 23rd Int. Conf. Mobile Human-Computer Interaction (MobileHCI ’21)*, New York, NY, USA: ACM, 2021, Art. no. 10, pp. 1–15, doi: [10.1145/3447526.3472049](https://doi.org/10.1145/3447526.3472049).
- [9] L. Findlater, K. Moffatt, J. E. Froehlich, M. Malu, and J. Zhang, “Comparing touchscreen and mouse input performance by people with and without upper body motor impairments,” in *Proc. 2017 CHI Conf. Human Factors Comput. Syst. (CHI ’17)*, New York, NY, USA: ACM, 2017, pp. 6056–6061, doi: [10.1145/3025453.3025603](https://doi.org/10.1145/3025453.3025603).
- [10] A.-I. Slean and R.-D. Vatavu, “Wearable interactions for users with motor impairments: Systematic review, inventory, and research implications,” in *Proc. 23rd Int. ACM SIGACCESS Conf. Comput. Accessibility (ASSETS ’21)*, New York, NY, USA: ACM, 2021, Art. no. 7, pp. 1–15, doi: [10.1145/3441852.3471212](https://doi.org/10.1145/3441852.3471212).
- [11] N. Caprani, N. E. O’Connor, and C. Gurrin, “Touch screens for the older user,” in *Assistive Technologies*. InTech, Mar. 16, 2012. doi: [10.5772/38302](https://doi.org/10.5772/38302).
- [12] J. Conradi, O. Busch, and T. Alexander, “Optimal touch button size for the use of mobile devices while walking,” in *Procedia Manuf.*, vol. 3, pp. 387–394, 2015, doi: [10.1016/j.promfg.2015.07.182](https://doi.org/10.1016/j.promfg.2015.07.182).
- [13] K. Lee, “Towards a working definition of designing generative user interfaces,” in *Companion Proc. 2025 ACM Designing Interactive Systems Conf. (DIS ’25 Companion)*, New York, NY, USA: ACM, 2025, pp. 489–495, doi: [10.1145/3715668.3736365](https://doi.org/10.1145/3715668.3736365).
- [14] L. Sun, “AI-assisted UI design: Enhancing efficiency and creativity through generative tools,” in *J. Comput. Technol. Appl. Math.*, vol. 3, no. 1, pp. 19–27, Jan. 2026, doi: [10.70393/6a6374616d.333638](https://doi.org/10.70393/6a6374616d.333638).
- [15] F. G. Deriba, M. Saqr, and M. Tukiainen, “Exploring the role of AI in enhancing accessibility in user interface design: A qualitative study of student experiences,” in *Proc. TEEM 2024*, R. Molina Carmona *et al.*, Eds. Singapore: Springer, 2025, doi: [10.1007/978-981-96-5658-5_44](https://doi.org/10.1007/978-981-96-5658-5_44).
- [16] N. Henze, E. Rukzio, and S. Boll, “100,000,000 taps: Analysis and improvement of touch performance in the large,” in *Proc. 13th Int. Conf. Human Computer Interaction with Mobile Devices and Services (MobileHCI ’11)*, New York, NY, USA: ACM, 2011, pp. 133–142, doi: [10.1145/2037373.2037395](https://doi.org/10.1145/2037373.2037395).
- [17] R.-D. Vatavu and O.-C. Ungurean, “Stroke-gesture input for people with motor impairments: Empirical results and research roadmap,” in *Proc. 2019 CHI Conf. Human Factors Comput. Syst. (CHI ’19)*, New York, NY, USA: ACM, 2019, Paper 215, pp. 1–14, doi: [10.1145/3290605.3300445](https://doi.org/10.1145/3290605.3300445).
- [18] O.-C. Ungurean, R.-D. Vatavu, L. A. Leiva, and R. Plamondon, “Gesture input for users with motor impairments on touchscreens: Empirical results based on the kinematic theory,” in *Extended Abstracts 2018 CHI Conf. Human Factors Comput. Syst. (CHI EA ’18)*, New York, NY, USA: ACM, 2018, pp. 1–6, doi: [10.1145/3170427.3188619](https://doi.org/10.1145/3170427.3188619).
- [19] M. Schmettow, “Sample size in usability studies,” in *Commun. ACM*, vol. 55, no. 4, pp. 64–70, Apr. 2012, doi: [10.1145/2133806.2133824](https://doi.org/10.1145/2133806.2133824).
- [20] K. Caine, “Local standards for sample size at CHI,” in *Proc. 2016 CHI Conf. Human Factors Comput. Syst. (CHI ’16)*, New York, NY, USA: ACM, 2016, pp. 981–992, doi: [10.1145/2858036.2858498](https://doi.org/10.1145/2858036.2858498).
- [21] R.-D. Vatavu, I. Petrariu, T. Horomnea, and O.-C. Ungurean, “Modeling touch input for users with motor impairments: Empirical insights into training size requirements,” in *Extended Abstracts 2026 CHI Conf. Human Factors Comput. Syst. (CHI EA ’26)*, New York, NY, USA: ACM, 2026, Art. no. 473, pp. 1–6, doi: [10.1145/3772363.3798487](https://doi.org/10.1145/3772363.3798487).
- [22] T. Cover and P. E. Hart, “Nearest neighbor pattern classification,” in *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: [10.1109/TIT.1967.1053964](https://doi.org/10.1109/TIT.1967.1053964).
- [23] H. A. Alfeilat *et al.*, “Effects of distance measure choice on k-nearest neighbor classifier performance: A review,” in *Big Data*, vol. 7, no. 4, 2019, doi: [10.1089/big.2018.0175](https://doi.org/10.1089/big.2018.0175).
- [24] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” in *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [25] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984, doi: [10.1201/9781315139470](https://doi.org/10.1201/9781315139470).
- [26] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000, doi: [10.1080/00401706.2000.10485983](https://doi.org/10.1080/00401706.2000.10485983).
- [27] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” in *Proc. 10th Int. Conf. Neural Information Processing Systems (NIPS ’96)*, Cambridge, MA, USA: MIT Press, 1996, pp. 155–161, doi: [10.5555/2998981.2999003](https://doi.org/10.5555/2998981.2999003).
- [28] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” in *Journal of Machine Learning Research*, vol. 7, no. 19, pp. 551–585, 2006, <http://jmlr.org/papers/v7/crammer06a.html>.
- [29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009, doi: [10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7).
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, 978-0-387-31073-2.
- [31] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed., CA, USA: O’Reilly Media, 2022.
- [32] J. E. Montandon, L. L. Silva, C. Politowski, G. E. Boussaidi and M. T. Valente, “Unboxing default argument breaking changes in Scikit Learn,” in *2023 IEEE 23rd Int. Working Conf. Source Code Analysis and Manipulation (SCAM)*, Bogotá, Colombia, 2023, pp. 209–219, doi: [10.1109/SCAM59687.2023.00030](https://doi.org/10.1109/SCAM59687.2023.00030).
- [33] H. J. P. Weerts, A. C. Mueller, and J. Vanschoren, “Importance of tuning hyperparameters of machine learning algorithms,” 2020. doi: [10.48550/arXiv.2007.07588](https://doi.org/10.48550/arXiv.2007.07588).
- [34] M. Stone, “Cross-validated choice and assessment of statistical predictions,” in *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974, doi: [10.1111/j.2517-6161.1974.tb00994.x](https://doi.org/10.1111/j.2517-6161.1974.tb00994.x).
- [35] A. Vamsikrishna and E. V. Gijo, “New techniques to perform cross-validation for time series models,” in *Oper. Res. Forum*, vol. 5, Art. no. 51, 2024, doi: [10.1007/s43069-024-00334-8](https://doi.org/10.1007/s43069-024-00334-8).
- [36] T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not,” in *Geosci. Model Dev.*, vol. 15, pp. 5481–5487, 2022, doi: [10.5194/gmd-15-5481-2022](https://doi.org/10.5194/gmd-15-5481-2022).
- [37] J. O. Wobbrock, K. Z. Gajos, S. K. Kane, and G. C. Vanderheiden, “Ability-based design,” in *Commun. ACM*, vol. 61, no. 6, pp. 62–71, 2018, doi: [10.1145/3148051](https://doi.org/10.1145/3148051).