TypeBoard: Identifying Unintentional Touch on Force-Sensitive Touchscreen Keyboard

ANONYMITY, address, Country

用户在触屏键盘上打字时，触摸屏幕即触发点击事件。因此，触屏用户不能像在物理键盘上一样通过触摸来align fingers positions[xx]，也不能将手指休息在键盘上，影响了触屏打字的效率和舒适度。在这篇论文中，我们提出了TypeBoard，一款带压力屏键盘上的防误触算法，我们也研究了用户使用TypeBoard时的打字行为。用户的打字行为和防误触能力之间是相互影响的，比如，在防误触的触屏键盘上，用户会更倾向于将手指休息在触碰上，造成更多的、更多样的非有意触摸点；而更多的、更多样的非有意触碰点会对防误触提出更高的要求。为此，我们通过迭代的数据采集和机器学习方法来设计了TypeBoard防误触算法。在一个使用TypeBoard写日记的评测实验中，用户的非有意触点个数占触点总数的xx.x%，我们的算法能在点击事件发生100 ms的时间内以xx.x%的准确率判断出其输入意图，而相关工作中基于压力和时间阈值的方法[xx]的识别准确率只有xx.x%，且必须在release以后做出判断。这份工作说明，第一，压敏触屏键盘有能力准确地、低延迟地防止用户休息、轻触等行为造成的误触。第二，用户在有效防误触的键盘上打字时，其用户行为会发生改变，比如手指休息的行为会显著增加。第三，防误触键盘和传统触碰键盘相比，能够防止用户疲劳，提高用户体验，且显著降低了输入任务的完成时间。

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Smart watch, text entry, touch input.

ACM Reference Format:
anonymity. 2018. TypeBoard: Identifying Unintentional Touch on Force-Sensitive Touchscreen Keyboard. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 16 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

随着平板电脑的普及[xx]，在平板电脑上进行文本输入的需求也在增长。平板电脑用户需要使用文本输入来完成搜索、笔记等简单的功能，有的用户甚至需要用平板电脑办公。触屏软键盘是平板电脑上通用的文本输入方案，然而，软键盘在输入效率（xx-WPM[xx] vs. xx-WPM[xx]）、疲劳程度[xx]和视觉注意力负担[xx]等诸多方面上与和物理键盘存在巨大的差异。在物理键盘上打字时，用户可以将手指休息在键盘上，这有两大优势：第一，用户可以将手指休息在键盘上，减轻疲劳；第二，用户可以通过触摸按键纹理来对齐他们的手指，从而实现盲打，降低视觉注意力的占用，大大提高输入效率。目前，用户不能将手指休息在软键盘上，因为这会导致误触。在篇论文中，我们提出了TypeBoard，一款软键盘上的防误触算法，使得用户可以将手指休息在触屏上。在TypeBoard的帮助下，我们有机会通过添加膜层[xx]或可变形触屏[xx]等方案给软键盘添加触觉纹理，从而支持软键盘上的盲打，弥补软硬键盘之间的差距。

在触屏键盘上区分打字点击和“误触”并不是我们的首创。在2013年，TapBoard就曾经提出将Tapping动作看作打字点击，而将其它触摸事件视为误触。Tapping的定义是“触摸时间低于xx毫秒，位移低于xx毫米的点击”。用户需要主动去适应TapBoard的技术方案，这存在着准确率低、影响用户自然性和舒适性

Author’s address: anonymity, anonymity@anonymity.com, address, P.O. Box 1212, Dublin, Ohio, Country, 43017-6221.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Association for Computing Machinery.
2476-1249/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

等等诸多问题，我们会在相关工作中详细描述。为了克服TapBoard的局限性，在这份工作中我们从用户的角度出发，将打字点击定义为“表达键入意图的点击”。特别地，我们希望理解用户在防误触触屏键盘上自然的打字行为，并根据用户的行为模式，来设计一款鲁棒的防误触触屏键盘。

“理解用户在防误触触屏键盘上自然的打字行为”是困难的，这涉及到人机交互研究领域中一个广泛存在的问题，即交互技术和用户行为之间可能是互相影响的。这一现象在防误触算法的设计上尤为突出：一方面，防误触算法的能力会显著影响用户的行为，举例而言，用户在防误触能力强的软键盘上会将手指休息在触屏上，带来更多更有挑战性的误触点；另一方面，我们对用户行为的理解又能帮助防误触算法的设计。然而，几乎所有相关论文都忽略了这一现象，有的论文在收集用户行为数据时没有给出反馈[xx,xx,xx]，其它论文给出了替代性的反馈[xx,xx,xx]。为了弥补这个缺陷，我们采用迭代的方式来设计TypeBoard，逐步理解用户的自然打字行为、设计与行为相符的防误触算法。我们共组织了三个用户实验，分别回答了以下三个研究问题：

(1) **RQ1):** 用户在一个想象中的防误触触屏键盘上的打字行为如何？我们组织实验一收集了用户在无反馈触摸板上打字的数据，用户想象该键盘能够完美地防止误触。由于触摸板没有反馈，用户不能真的打字，而是想象字母上屏了。用户完成了多种文本输入任务，并人工标注了数据。实验一共采集了xx个数据点，其中误触点占xx%，远大于我们在正常触屏上的误触数量。我们分析了用户的行为，并开发了针对性算法，识别误触的准确率达到了xx.x%，延迟为手指落下后xx毫秒。作为比较，TapBoard[xx]在该数据集上的准确率仅为xx.x%，且只能在手指抬起时识别。
(2) **RQ2):** 用户在防误触触屏键盘上的打字行为如何？在实验一以后，我们得到了初步的防误触算法，我们组织实验二收集了用户在该防误触键盘上打字的数据。在实验二中，用户能够键入字母，同时得到声音反馈。用户完成了多种文本输入任务，并人工标注了数据。共有25名用户参与了实验二，他们被分成了五组。在每五个用户完成实验之后，实验者都会更新数据集，添加必要的特征向量，重新训练防误触算法。最终，实验二共采集到xx个数据点，其中误触误触点占xx%，显著高于实验一的误触点数量，这暗示真正的防误触算法会提高用户对触屏的信任感。在该数据集上，我们的算法的识别准确率达到了xx.x%，进一步拉大了与TaoBoard（xx.x%）的差距。
(3) **RQ3):** 防误触键盘对用户体验和效率有和影响？防误触键盘上加上纹路有何影响？ 我们组织了实验三，在多个输入任务下对比了TypeBoard和普通触屏键盘的用户体验和输入效率。结果表明，TypeBoard在写日记、填问卷等任务下显著提升了用户体验，降低了疲劳，而在誊写任务下和普通触屏键盘没有差异。实验三同时对比了在有无纹理反馈两种设置下用户的行为规律，和TypeBoard性能变化，结果发现，纹理反馈使得用户更频繁地将手指放置在触摸屏上，误触点数大大增加，在这一极端的数据集下，我们的算法准确率为xx.x%，而baseline的准确率仅为xx.x%。纹理反馈大大提升了用户的输入效率，观察实验视频我们发现，这一提升可能是因为防误触算法+纹理反馈合力使能了用户在触屏键盘上的盲打。

这份工作有三个贡献点：第一，TypeBoard准确地、低延迟地区分了触屏打字时的typing和误触。第二，在TypeBoard的支持下，我们理解并总结了用户在防误触触屏键盘上的输入行为，我们公开了该数据集。第三，我们通过评测实验证明，TypeBoard与传统触屏键盘相比，提高了输入效率，降低了用户疲劳程度。

2 RELATED WORK

2.1 Unintentional Touch Rejection on Touchscreen

Touch is the main input channel on touchscreen devices such as smartphone, tablet and tabletop, but not all contacts on the touchscreen are intended to trigger a digital response. Those touches that do not contribute to any interaction goal are known as unintentional touches [26], or namely accidental/unwanted touches [17, 18]. Since unintentional touches trigger unwanted interaction, the user's on-going behavior will be interrupted by the unintentional touch [xx]. Moreover, the user needs to spend extra time to cancel the accidentally triggered response [xx], which affects the efficiency [xx] and naturalness [xx] of the interaction [2, 26].

However, unintentional touch is inevitable in touchscreen interaction. For example, the thenar eminence on the human hand will constantly contact the touchscreen during the daily use of smartphones [xx]. Fortunately, we can identify and filter out these unintentional touches by software techniques. In the literature, the methods of preventing unintentional touches has been extensively studied. We compare existing work in two aspects: (1) the use scenario, and (2) the sensor.

2.1.1 Unintentional Touch Rejection over use scenarios. The definition of unintentional touch varies in different use scenarios. When the application is not limited, unintentional touches refer to those touches that do not contribute to any interaction goal [26]. The boundary of intentional and unintentional touches will be more clear in specific task. For examples, in the text entry task, unintentional touches are those touches that do not intended to entry words [xx].

A few studies [18, 26] and a large amount of patents [5, 6, 11, 20, 21] attempt to identify unintentional touch over applications. Metero et al. presented guidelines to reduce the amount of unintentional touches on smartphones [18]. Their filtering criteria rejected 79.6% of unintentional touches. Xu et al. identify and filter out unintentional touches on interactive tabletop using gaze direction, head orientation and screen contact data [26]. The accuracy was 91.3%. These approaches suffered from low recognition rate.

In the literature, more studies were conducted to identify unintentional touch in specific scenarios. TapBoard and TapBoard2 discussed this issue in text entry tasks [12, 13]. TapBoard regarded short-term tapping actions as keystrokes and other contacts as unintentional touches. The system reported a keystroke when the touch duration is shorter than 450 ms and the touch movement is shorter than 15 mm. Users adapted their behaviors to these thresholds, so that TapBoard achieved an accuracy of roughly 97%. Based on TapBoard, TapBoard2 was able to disambiguate typing and pointing actions with an accuracy of greater than 95%. While inking on tablets, unintentional touch resulted in a great effect on user behavior and was one of the most prominent features identified as problematic by participants [2], e.g., users were forced to write in an uncomfortable position to avoid the ‘palm touch’ screen. Several studies were proposed to reject unintentional touch in pen and tablet interaction [3, 9, 22]. Schwarz et al. achieved the best performance [22] by leveraging spatiotemporal touch features, reducing accidental palm inputs to 0.016 per pen stroke, while correctly passing 98% of stylus inputs. In smartphone interaction, palm touches are often considered as unintentional touches. PalmTouch used these "unintentional touches" as intentional input methods [14], such as a shortcut. PalmTouch differentiated between finger and palm touch with an accuracy of 99.53% in realistic scenarios. GestureOn enabled gesture shortcuts in the standby mode by which a user can draw a gesture on the touchscreen before the screen is turned on [17]. GestureOn acquired 98.2% precision and 97.6% recall on detecting gestures from accidental touches. The studies above explored the feasibility of rejecting unintentional touch in specific scenarios. Their performances were generally high.

2.1.2 Unintentional Touch Rejection over sensors. A mass of studies have been conducted to recognize unintentional input on touchscreen devices, including smartphones [14, 15, 17, 18], tablets [3, 11–13, 22] and tabletops [26]. Most techniques leverage spatiotemporal features of touchpoints [11–13, 18, 22] and capacitive images to identify unintentional touches [3, 14]. Metero et al. explored the feasibility of rejecting unintentional touch on smartphones using touchpoint patterns [18]. They analyzed user behavior in three typical tasks including swipe interactions in the home view, phone call interaction, and general device handling. The authors proposed filtering criteria such as touch duration, position and trajectory pattern that rejected 79.6% of unintentional touches whilst rejecting 0.8% of intentional touches. Schwarz el al. presented a probabilistic touch filtering approach that distinguish between legitimate stylus with palm touches on tablet computers [22]. The method extracted features from touchpoints and used the decision forest model, reducing accidental palm inputs to 0.016 per pen stroke, while correctly passing 97.9% of stylus inputs. PalmTouch [14] is an additional input modality that differentiates between touches of fingers and the palm. The intended palm touch supports different use cases, including the use as a shortcut and for improving reachability. PalmTouch used the raw capacitive image of the touchscreen as input and used Convolutional Neural Network (CNN) as the method, resulting in an accuracy of 99.53% in realistic scenarios. The above approaches rely on built-in sensors, and can be immediately applied to most of the existing smartphones and tablet computers. As a trade-off, they are limited by low recognition accuracy or fixed application scenarios.

Some related techniques enhanced the sensing ability to improve the performance of rejecting unintentional touches [8, 15, 17, 26]. GestureOn [17] distinguishes between intended gesture input with unintentional touches in the standby mode of smartphones. The user can trigger gesture shortcuts before the screen is turned on. GestureOn used most of the built-in sensors on the smartphone including proximity sensor, light sensor, IR sensors and Inertial Measurement Unit (IMU). The system also leverage the pressure associated with the touch event, which is not popularized on smartphones yet. Based on sensor fusion, GestureOn acquired 98.2% precision and 97.6% recall on detecting gestures from accidental touches. Xu et al. leveraged gaze direction, head orientation and screen contact data to identify and filter out unintentional touches on interactive tabletop [26]. Result showed that the patterns of gaze direction and head orientation improved the accuracy of identifying unintentional touches by 4.3%, reaching 91.3%. The above approaches used additional sensors, and unsurprisingly improved the recognition rate.

2.1.3 Summary. As table xx shows, we proposed a classification of unintentional touch rejection methods into two broad categories: *universal method* and *specific method*. We define *universal methods* as those techniques that identifies and filters out unintentional touches over different scenarios. For *universal methods*, unintentional touches refer to those touches that do not contribute to any interaction goal. We define *specific methods* as those techniques that identifies unintentional touches in specific scenarios. For example, unintentional touches in the text entry task are those touches that do not represent typing intent. As table xx shows, the recognition rate of *specific methods* are generally higher that the performance of *universal methods*. We argue that this gap is significant. As the performances of *universal methods* were not higher then 92%, the lowest achievable error rate (Bayes error rate [24]) might be high. *Universal methods* do not significantly change user's behavior, i.e., users were forced to perform unnatural and uncomfortable actions to avoid unintentional touches [2] [xx]. For comparison, the performances of *specific methods* are much higher. Users are willing to change their behavior to acquire benefits from the techniques [12, 13].

Beside use scenario, the sensing ability of the devices also affect the recognition rate of unintentional touch. Several studies have shown that additional input channel such as xx, xx and xx significantly improve the accuracy. In this work, we used a device with strong sensing ability (force-sensitive touchscreen) to identify unintentional touch in a specific task (text entry). There was no doubt that we were able to reach a high recognition rate. In this situation, we were interested in a deeper question, which is the interrelationship between technique and user behavior. On the one hand, a perfect technique of rejecting unintentional touch will enable natural and relaxing user behavior on the touchscreen. On the other hand, a perfect technique should adapt to the natural user behavior. To our knowledge, previous studies did not analyze natural user behavior on a unintentional touch rejecting touchscreen. They either collected user data in a system with no feedback [xx,xx,xx] or emulational feedback [xx]. In this paper, we introduce a iterative process to reveal the interrelationship between unintentional touch rejection technique and user behavior.

TapBoard [13] was similar to our work. We both investigate unintentional touch rejection on touchscreen keyboard. However, TapBoard used thresholds to distinguish between typing and unintentional touch, thus users need to adapt their behaviors to the technique. This could not provide natural and relaxing user experience [xx, xx]. Moreover, TapBoard suffered from modest accuracy (97%) and high recognition delay (recognized at the touch up moment). In this paper, we cover TapBoard's shortage by considering the interrelationship between technique and user behavior.

	universal method	specific method
off-the-shelf sensors	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]
	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]
	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]
additional sensors	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]
	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]
	xxx (xx.x%) [xx]	xxx (xx.x%) [xx]

Table 1. Classification of unintentional touch rejection techniques.

2.2 Benefits from Unintentional Touch Rejection

Since unintentional touch affects the efficiency and naturalness of the touch interaction, a straightforward benefit from unintentional touch rejection is to improve the user experience. Beside this, we summarize other advantages of unintentional touch rejection as follows.

2.2.1 Bridge the Gap Between Soft Keyboard and Physical Keyboard. Though touchscreen devices are getting popular in the past ten years [xx], users still prefer to use personal computers and physical keyboards in the office work [xx]. This is because physical keyboards are better than touchscreen keyboards in usability [xx] and efficiency (xx-WPM vs. xx-WPM [xx]). Users can rest their fingers on physical keyboards, which is an important factor in the usability of a keyboard [13]. First, users avoid fatigue by resting their fingers in a long text entry task. Second, users align their fingers by touching the tactile landmarks on physical keyboards. This alignment behavior enable efficient touch type.

Our work support unintentional touch rejection on touchscreens, that is, users are able to rest their fingers on the touchscreen. This improvements has three advantages. The first is avoiding fatigue. Second, our studies show that the resting behavior improved the accuracy of touch position in each keystroke, because the users' fingers got closer to the desired keys. Third, we added stickers on the touchscreen keyboard to provide tactile landmarks and found it significantly improve the accuracy and efficiency. In real use, deformable touchscreen surfaces [1, 4, 16] and tangible keyboard objects on the touchscreen [23, 25] can provide such tactile landmarks. Thus, unintentional touch rejection does bridge the gap between soft keyboard and physical keyboard.

2.2.2 Leveraging Unintentional Touch as Input Channels. The goal of our work is filtering out unintentional touches on the touchscreen. However, there are some literature that tried to "use" unintentional touches as situational awareness contexts [xx] or explicit interactions [xx]. For situational awareness, Zhang et al. leveraged unintentional touches such as the palm touch to augment pen and touch interactions, e.g., providing "Palm Tools" that teleport to a hand-centric location when the user plants their palm on the screen while drawing [27]. For explicit interactions, Matulic el at. enriched tabletop interaction by considering whole hands as input [19]. The algorithm detects seven different contact shapes with 91% accuracy and can be used to trigger, parameterise, and dynamically control menu and tool widgets. PalmTouch is an additional input modality that differentiates between touches of fingers and the palm [14]. PalmTouch can be used as a shortcut for improving reachability. TapBoard2 [12] distinguishes typing from pointing using a threshold method, thereby unifying the keyboard and mouse control spaces. This depresses the burden of frequently switching between devices [7].

3 STUDY 1: USER BEHAVIOR ON IMAGINARY TYPEBOARD

In this study, we collected data from participants' typing actions on a force-sensitive touchpad. The motivation was to investigate users' typing behavior on an imaginary touchscreen keyboard that can "perfectly reject unintentional touch", based on which we can design the algorithm of rejecting unintentional touch. Because (incorrect) feedback will affect users' behaviors, in this study, participants typed on a touchpad without any feedback. Participants could not enter words by typing on the touchpad, instead they imagined that the desired words are entered. Besides, participants need to adapts their typing behaviors according to the imagination that the keyboard can perfectly reject unintentional touch.

3.1 Participants

We recruited 16 participants from the campus (aged from xx to xx, M = xx.xx, SD = xx.xx, xx females). All the participants were right handed and native Chinese speaker. They have used software keyboards on smartphones for not less than xx years. xx of the 16 participants were familiar with software keyboards on tablets.

3.2 Design and Procedure

Figure xx illustrate the experimental setting. There was a Morph Sensel [xx] force-sensitive touchpad on the desk. We placed a windows surface tablet on the touchpad, covering less than half of the pad. We drew a QWERTY layout on the touchpad using highlighter pens. This combination of touchpad and table took place of the force-sensitive touchscreen, which is expected to be commercialized in the future. Participants filled in a Microsoft Word document to complete the experimental tasks. They touched on the tablet for pointing, and typed on the touchpad to "enter words". The system recorded every touches and the screencast during the experiment.

The experiment included four sessions of text entry tasks: (1) filling in personal information, (2) describing personal tastes, (3) open-book examination, and (4) picture writing. We counterbalanced the order of the four tasks using a balanced latin square. We did not include transcription as a task as many other studies of text entry do [xx,xx,xx], because our pilot study showed that participants seldom rested their fingers on the touchpad in a transcription task, resulting in low efficiency of obtaining unintentional touches. The details of the four tasks are as follows:

(1) **Filling in personal information:** In the task document, there was a table of xx question about personal information, such as name, gender, and so on. Participant direct touched on the tablet to select what to fill, and typed on the touchpad the "enter words". To protect users' privacy, users were allowed to fill in fake information. This assignment represented those text entry tasks that require frequently switching between pointing and typing.
(2) **Describing personal tastes:** There was another table of xx question about personal tastes, such as "the favorite city", "the favorite fruit", and so on. Participant touched on the tablet for pointing and pressed on the touchpad for typing, as if they were doing office work. Participant were allowed to fill in fake information.
(3) **Open-book examination:** The "exam" consisted of five hard questions such as "What is the 50th element of the periodic table?". Participants could hardly know the answer, so that they needed to use the search engine. The system recorded both the behaviors of answering questions and searching in the Internet. Because the participants could not enter words in the search engine, they needed to speak out when typing, so that the experimenter could replaced them to enter the words. This assignment represented those text entry tasks that require using the search engine.
(4) **Picture writing:** There was a picture (as figure xx shown) in the task document. Participants needed to describe the picture in five sentences and wrote down the story in the document. We asked participants to speak out while typing, so that the experimenter could replaced the participants to enter the words. This assignment represent the text entry tasks that the users need to think while writing.

Before the experiment started, the participant had five minutes to familiarize himself with the tasks and the requirements of the experiment. During the warm-up phase, the participant "typed" on the touchpad freely, while the experimenter reminded the user of two points. First, the keyboard did not provide any feedback. Participants could not enter words, but imaged that they entered words. Because the users' first language are Chinese, which involved word selection in the text entry method, users assumed that the desired word is always the first one of the candidate words. Second, users needed to imagine that the keyboard can perfectly prevent unintentional touches, and adjust their behavior according to this assumption. For example, they could rest their fingers on the keyboard while thinking. This is not mandatory. Participants could make choices as they wished.

After finishing each session of task, the participant labeled the data through an interactive program. The program showed the capacitive images of touchpad and the screencast of tablet at the same time. As figure xx shows, there were some red points on the capacitive images that showed the touchpoints reported by the touchpad. Participants labeled the intended touches as green points. Because participants got context information from the screencast, they were able to identify most intentional touches. If participants were not sure, they could label the touchpoint as a blue point to remove the data. In average, participants spent five minutes to finish the text entry tasks and spent 45 minutes to label the data. Participants rested for five minutes between two sessions to avoid fatigue. The study was generally completed within 70 minutes.

3.3 Apparatus

As figure xx shows, we placed a Windows surface tablet and a Morph Sensel force-sensitive touchpad [xx] together to simulate a tablet computer that contains force-sensitive touchscreen. The Sensel Morph is a multi-touch and force-sensitive touchpad, which sense the position and the pressure level of touches. The Sensel Morph contains 185 x 105 sensor elements ("sensels") at a 1.25mm pitch. Each contact can sense approximately 30000 levels, ranged from 5g to 5kg. The upper limit of frequency is 125Hz (8ms latency), while we slowed it down to 50Hz to fetch stable data. The Sensel Morph provides capacitive images and touchpoint information including position, timestamp, touch area, pressure level and shape. The recognition of touchpoint is sensitive that almost every contacts are reported as touchpoints, so in this paper, we identified unintentional touches among reported touchpoints, while did not consider missing touches by the Morph Sensel.

【图：实验设置，特别是平板电脑+压力触控板=压敏平板电脑】

The size of sensing area on the Morph Sensel is 240mm x 138mm. We used highlighter pen to draw a Qwerty layout on the touchpad as shown in figure xx. The width of the Morph Sensel (240mm) is a little shorter than the width of the keyboard on a 15 inches MacBook (270mm). We removed some keys that are less frequently used such as square brackets and semicolon, so that the Qwerty layout could be placed in the Morph Sensel, while the size of each key reminded the same as Macbook. Because the Qwerty layout is changeable in software keyboard, we did not used the layout as prior knowledge in the recognition of unintentional touch.

The tablet we used in the experiment was Windows surface xx, with ixx Intel Core Processor, xx-core. The data collecting program ran on the tablet at a stable frequency of 50 FPS.

3.4 Result

The dataset contains 12659 touches, excluding the ambiguous touches (0.18%) in the labeling. 67.5% of the data were positive samples (intentional touches), while 32.5% were negative samples (unintentional touches). Because some participants misunderstood the concept of "unintentional touches", there might be mislabeled data points. We trained a simple machine learning model (version 1) to process the data. If there were a large between labels and predicting results, we asked the participants to relabel the suspicious data points.

3.4.1 *Model version 1: naive model and data processing.*

We first trained a simple and straightforward model (version 1) for data processing. We sampled the first five frames of each touch. If the duration of a touch is shorter than five frames, the whole touch was regarded as a sample. As table xx (V1) shows, we extracted features from the samples as follows: for the contact area, ellipticity, displacement, force and intensity over frames, we calculated their temporal feature including maximum, minimum, mean, skewness and kurtosis. Then, we concatenated these values to obtain a feature of 25 dimensions and trained an SVM binary classifier. Positive and negative samples were balanced in weight.

We used this model to simulate the dataset and observed. We found that some participants misunderstood the concept of "unintentional touches", for example, regarding wrong inputs as unintentional touches. We asked the participants to relabel the suspicious data points through e-mail. Finally, we had xx data points, including xx.x% positive samples and xx.x% negative samples.

Leave one out cross-validation shows that the accuracy of model version 1 was 96.86% (SD=4.17%). For comparison, if we did not have the force signal (i.e., the regular touchscreen devices), the accuracy would reduce to 92.30% (SD=4.88%). The result shows that the force signal is important for unintentional touch rejection (F,p).

Version	Criteria	Feature	Introduction	Accuracy
V1	The current touch (no force sensor)	Contact area	The contact area in pixels.	96.86% (SD=4.17%)
		Ellipticity	The contact region is fitted as an ellipse. The ellipticity is the ratio of the minor axis to the major axis.	
		Displacement	The distance to the starting location.	
	The current touch (with force sensor)	Force	The contact force in grams.	
		Intensity	The ratio of the force to the contact area.	
V2	The proportion of the current touch	Force fraction	The ratio of the force to the total force of all contacts.	98.59% (SD=1.46%)
		Intensity fraction	The ratio of the intensity to the total intensity of all	
		Area fraction	The ratio of the contact area to the total contact area of the full screen.	
V3	Location of the current touch	Distance to edges	The minimum distance to one of the three edges, including the bottom and the two flanks.	**99.07% (SD=0.71%)**
		Distance to corners	The minimum distance to one of the two corners, including the bottom left corner and the bottom right corner.	
	Relationship to recent/nearby touches	Recent touches	The relationships between the current touch and the last five touches in five seconds. The relationships include (1) the interval betwen their start times, (2) the average distance, and (3-5) the ratios in terms of contact area, force and intensity. If there are less than five touches, complete the feature vector with zeros.	
		Nearby touches	The relationships between the current touch and the five nearest touches within the liftcycle of the current touch. If there are less than five touches, complete the featue vector with zeros.	

Fig. 1. The features we fed into the SVM model and the accuracy among model versions.

3.4.2 Model version 2: filtering out multiple fingers resting. Observation showed that multiple fingers resting was the most frequent unintentional touches, where users rested more than two fingers on the screen at the same time. There were more than x0% of multiple finger resting among all unintentional touches. Thus, we add a series of features in the model version 2 to deal with this user behavior. A new criterion was added in the model, namely "the proportion of this touch among all touches". The features in details can be found in table xx (V2). Leave one out cross-validation shows that the accuracy increased to 98.59%.

3.4.3 Model version 3: understand user behavior and improve the model. To understand the user behavior and further improve the model, we analyzed the fail cases of the model version 2. As table xx shows, the error rate of model version 2 was 1.41%, including xx.x% false triggering touchpoints and xx.x% unrecognized touchpoints. We classified the fail cases into 16 categories. We counted their frequencies, analyzed the reasons, and gave possible solutions.

For the fail cases with a white background (xx.x%) in table xx, the researchers could predict correctly without watching the experiment screencast. That is, the machine should have predicted correctly if it is as smart as a human. For these cases, we proposed features to improve the model. Here are some examples:

(1) **EG1):** *Pisiform.* As figure xx.x shows, the pisiform is a small bone found in the proximal row of the wrist (carpus), while the thenar eminence is the group of muscles on the palm at the base of the thumb. Both the posiform and the thenar eminence would contact the touchscreen when a user is typing. However, the touches caused by the posiform is usually heavy and intensive, which is easy to confused with intentional touches. Furtunately, these touches are frequently in the bottom left and the bottom right corners. Thus, the distances to the corners could be good features to reject these unintentional touches.
(2) **EG2):** *Continuous touches.* When a user continuously typed on the same key (e.g., the delete key), the follow-up touches might be lighter. These touches are lighter than other intentional touches, so they are likely to be recognized as unintentional touches. Adding information of recent touches could help to correct this fail case.
(3) **EG3):** *One hand-typing.* Some participants typed with one hand, while resting the other hand on the touchsreen. In this situation, the model version 2 might think that all the touches together was "multiple fingers resting", and identified the typing as an unintentional touch. To correct this fail case, we added information of nearby touches, because the typing finger is usually far away from the resting fingers.

【图：大鱼际和小鱼际，还有其它需要图解的错例。】

However, there are xx.x% of the fail cases (with a gray background in table xx), that humans (the researchers) could not predict without watching the experiment screencast. We deemed that these fail cases were inevitable, because the machine can not know what the user is going to enter in advanced. Here are some examples. First, sometimes the participants rested one finger on the touchscreen heavily, which is indistinguishable with a intentional touch. Second, some participants performed a very light touch during entering a word, which is similar to an unintentional touch. That is, the accuracy of the model has an certain upper limit (roughly xx.x% in this dataset), while our goal is to approach it.

According to the user behaviors summarized in table xx, we added two criteria in the model training. The first criterion is the location of touch, including the minimum distances to the edges and to the corners. The model did not leverage the priori knowledge of the keyboard layout, because the layout of a software keyboard is changeable, while we wanted a universal model. The second criterion is the relationships between the current touch and the recent/nearby touches.The features in details can be found in the table. The model version 3 used all the features listed in table xx. We concentrated these features to form a vector of 100 dimensions and trained an SVM binary classifier. Leave one out cross-validation showed that the accuracy was 99.07%. The error rate was 0.93%, including xx.x% false triggering and xx.x% unrecognized touchpoints.

We used previous work as baseline [13], where every touches that last for more than 450 ms or move father than 15 mm are identified as unintentional touches. We adjusted the thresholds to xx ms and xx mm so that the baseline performed the best on our dataset. Our result (99.07%) surpassed the baseline (xx.xx%). In real use, our system calls the classifier once a touch has lasted for five frames or is released in advanced. The system reports the touchpoint if the prediction result is positive. The delay of our method is 100 ms. For comparison, the baseline can only judge a touch when it is released.

We have three conclusions in this experiments. First, participants were willing to rest their fingers on the touchscreen that can reject unintentional touches in imaginary. Hereby, we proposed a hypothesis to be verified, where users are willing to rest their fingers on a real TypeBoard. Second, the force signal is important for rejecting

Cases	Introduction	Fail cases	What features can improve the model?
False Reported Touchpoint			
Pisiform	The pisiform refers to the bone on the anteromedial side of the wrist. It usually contact the screen while typing.	23	Touchpoint location, e.g., is the location near the pisiform?
Thenar eminence	The thenar eminence is the group of muscles on the palm at the base of thumb. It usually contact the screen while typing.	2	Touchpoint location, e.g., is the location near the thenar eminence?
Repeated reporting (spatial)	The system recognized a touch as two touchpoints when the contact area is too large.	12	Info. of nearby touchpoints, e.g., is this touch near other touches?
Repeated reporing (temporal)	The system recognized a touch as two touchpoints when the touch is nearly released midway.	3	Info. of recent touchpoints, e.g., is there a touch just now?
Edge touch	Users trigger unintentional edge touch when adjusting the placement of devices.	7	Touchpoint location, e.g., is the location near the edge?
Two fingers resting	Users have two fingers rested on the touchscreen.	3	Info. of nearby touchpoints, e.g., is this touch near other touches?
Extra touchpoint (light)	Users trigger extra touchpoints while entering a word. These unintentional touches are usually lighter than the recent intentional touches.	9	Info. of recent touchpoints, e,g., is this touch lighter than recent touches?
Slide	A touchpoint with large displacement is less likely to be an intended keystroke.	7	Touchpoint displacement.
One finger resting	Users have one finger rested on the touchsreen. This is similar to an intended touch.	9	No solution.
Extra touchpoint (heavy)	Users trigger extra touchpoints while entering a word. This is similar to an intended touch if the touch is heavy.	15	No solution.
False Unreported Touchpoint			
Continuous touches	When a user continuously type on the same key (e.g., the delete key), the follow-up touches may getting lighter.	13	Info. of recent touchpoints, e.g., is this one of the continuous touches?
Rollover-typing	The next key is pressed before the previous is released [2018-Observation].	12	info. of recent touchpoints.
One-hand typing	Users type with one hand, while the other hand is resting on the touchscreen. This case is similar to multiple fingers resting.	6	info. of nearby touchpoints, e.g., is this one hand typing?
Palm touch	Users rest the palm on the screen and type. If the palm is detected as multiple fingers, this case is similar to multiple fingers resting.	5	info. of nearby touchpoins, e.g., is this touch near other touches?
Light touch	The very light but intended touch, which seems like an unintentional touch.	46	No solution.
Small contact area	The intended touch with a very small contact area. This is similar to an unintentional touch.	6	No solution.

Fig. 2. Fail cases of the model version 2. Noticed that the "Fail cases" is the number of fail cases of the model, but not a counting for the user behavior. xxx.

unintentional touches in the text entry task. Third, most unintentional touches (>99%) can be identified by spatial-temporal features of the signals on a force-sensitive touchsreen keyboard.

3.5 Discussion

3.5.1 Why sample five frames in each touch? There is a trade-off between the amount of sampling frames and the accuracy. The more data we sample in each touch, the more accuracy the prediction is. However, a long sampling window means a large delay, which affects the user experience. We needed to strike a balance. Five frames of sampling results in an acceptable prediction accuracy (99.07%), meanwhile the delay of 100ms is not perceivable in the touching task [xx].

3.5.2 What we found about user behavior? Here are some conclusions. First, the unintentional touches mainly contains three categories, including multiple fingers resting (xx.x%), palm touches (xx.x%) and others (mostly mistaken touches during input, xx.x%). Second, the text entry task has significant effect on the amount of unintentional touches (F, p). Bonferroni-corrected post-hoc tests showed significant differences between the following task pairs: xx-xx (p<), xx-xx (p<), and xx-xx (p<). Thus, we suggest that studies of unintentional touch should consider the variety of task. We will discuss the user behavior in details in the next study, because it is more meaningful to analyze the user behavior on a real TypeBoard.

4 STUDY 2: USER BEHAVIOR ON THE TYPEBOARD

In this study, we obtained users' typing data on the TypeBoard, a touchscreen keyboard with unintentional touch rejection. Participants used the TypeBoard developed in study one. The motivation was to investigate users' typing behavior, based on which we can further improve the identification of unintentional touch.

4.1 Participants

We recruited 16 participants from the local campus (aged from xx to xx, M = xx.xx, SD = xx.xx, xx femakes). All the participants were righted handed and did not took part in the first study. They have used software keyboards on smartphones for not less than xx years. xx of the 20 participants were familiar with software keyboards on tablets.

4.2 Design and Procedure

As figure xx shows, the experimental devices were the same as those in study one, including a Windows surface tablet and a Morph Sensel force-sensitive touchpad. The working frequency was 50 FPS. There were four sessions of text entry tasks, which also reminded the same as study one. The differences in study two are as follows. First, participants could actually enter words by typing on the touchpad. Second, the system supported unintentional touch rejection. Intentional touches would trigger keystrokes and audio feedback, while unintentional touches were ignored.

Before the experiment, participants warmed up by a ten-minute daily writing. Because no participant had experience with typing on a keyboard with unintentional touch rejection, we reminded the participants that they could rest their fingers on the keyboard while thinking. The resting behavior was not mandatory. Participants could decide to rest their fingers or not according to the task and their preference.

After finishing each session, participants labeled the data through the interactive program introduced in study one. During the labeling phrase, we compared participants' labels with the model predictions. When the two results were different, we immediately recorded and analyzed the data point. If the experimenter could not give reasons for the difference, he discussed with the participant. In average, participants spent 10 minutes to finish the text entry tasks and spent one hour to label the data. It took more time than study one, because participants needed to enter correct words in study two. Participants rested for five minute between two sessions to avoid fatigue. The study was generally completed within 90 minutes.

4.3 Result

实验二共收集了xx个数据点，已经排除了xx.x%的用户无法区分的数据点。在经过了和实验一相同的label确认流程之后，数据集中共包含xx.x%的正例和xx.x%的负例。实验一中总结出来的模型在实验二

的测试集下，预测准确率为98.05%(SD=1.51%)，有xx.x%的漏报和xx.x%的误触发。也就是说，在实验二的实验过程中，用户每有100次有意点击中，平均有xx.x次漏报和xx.x次的误触点。

观察发现，在实验二采集到的数据中，（与实验一相比）没有出现新的误触类型，而各种误触的比例发生了变化。在实验二的误触数据中，多指休息、手掌误触和其它误触的比例为xx.x%，xx.x%和xx.x%。方差分析显示，实验二中多指休息和手掌误触的数据比例都减少了(F,p;F,p)，而其它误触则显著增多(F,p;F,p)。为了给后续的工作提供指导，我们人工分析了实验二中所有误触的类型和其出现频次，如表格xx.x所示。在实验二的数据中，容易引发未识别的行为也发生变化。一个典型的例子是，实验二中用户连续点击删除键或方向右键（选词）的行为显著增多，这些属于continous touches，即follow-up touches may getting lighter。以上观察说明，实验二所采集到的用户行为与实验一所采集到的用户行为有着许多的不同点，它更接近用户在一个真正可以防误触的键盘上打字的自然行为。

【表格：用户的误触行为分类和出现频次】

因此，我们需要将预测算法的训练集替换成实验二所采集到的数据集，重新训练机器学习模型。果然，Leave-one-out验证发现算法的识别准确率提高至98.88%（SD=0.73%），且这一提高是显著的（F,p）。值得注意的是，我们仅仅替换了训练集（而非新增了训练集），这说明此提高来自数据对用户真实行为的契合度，而非数据量的增加。

4.4 Discussion

（1）迭代实验能否为用户行为探索和算法开发带来优势?

与相关工作相比，这篇论文的一个亮点是采用了迭代的方法来求解问题，即先开发一个版本的技术，采集用户在该技术下的数据，然后在新的数据集下分析用户行为和改进算法。由于技术和用户的行为之间往往是一个蛋鸡悖论，许多相关工作都仅仅在无反馈的设备（like our study one）上进行数据采集[xx,xx,xx]，或者是通过物理操作来模拟[xx,xx]。现在的问题是，我们所采取的迭代式方案是否能带来优势?

我们认为答案是肯定的，迭代的方法在一部分人机交互技术的研发过程中是值得采用的。以我们这份工作为例，我们在迭代实验中采集到的数据，与无反馈实验中采集到的数据相比，其用户行为的分布存在着很大的差异，且使用迭代实验的数据进行模型训练，也能显著提高算法的性能。因此，我们争辩我们的迭代方法值得更多的关注。

（2）任务的多样性在防误触算法的设计中是否重要?

统计分析表明，实验任务对用户的误触行为有显著的影响，首先是用户在不同任务下误触的次数就有显著差异，用户在xxx等比较xxx的任务下休息比较多，在xxx等任务下休息比较少；用户在xxx任务下会出现更多的输入间误触，而在xxx等比较xxx的任务下输入间误触比较少。上述结论说明，多任务的实验设置对误触行为的分析和算法构造很有必要。

基于以上结果，我们认为任务的多样性在防误触算法的研究和评测中需要被考虑到，这一问题在相关工作中经常被忽视[xx]。特别的，尽管誊写是文本输入法文献中最常用的实验任务，但是采用誊写实验来研究或评测误触算法[xx]是错误的，因为我们在预实验中发现，誊写任务中用户的误触行为非常少，这一点将在下个实验中得到验证。

（3）不同硬件对识别的影响。

现实中大多数触摸屏设备还没有压力传感器，有的设备（如MacBook的force touch trackpad）则在四个角上有压力传感器。为了给防误触触屏提供硬件设计指导，我们对比了我们的算法在三种情况下识别准确率，三种情况分别是：（1）capactive-only：只有电容信号；我们将算法中涉及到压力信号的特征维度都阉割掉来训练机器学习模型。（2）four-force-sensor：四个角上有压力传感器，根据实验采集到的高精度压力信号，我们可以根据基础的物理知识模拟四角压力传感器的数据。我们在情况1的基础上，直接加上四个角压力信号的时域特征。（3）force-enabled：既有电容信号，又有压力信号，这也就是我们论文中所述的配置。

【图：三种硬件设置下算法的识别准确率】

如上图所示是这三种情况下算法的识别准确率。值得注意的是，四个角上有传感器的识别准确率为xx.xx%，这说明，四个角上安装传感器是一个准确率不错、且目前为止容易实现、成本较低的实现方法。

5 STUDY 3: EVALUATION ON TYPEBOARD

The motivations of study three were two-fold. First, we evaluated the performance and user experience of TypeBoard on different text entry tasks. The baseline was the software keyboard without unintentional touch rejection. Second, as we introduced in related work, tactile landmarks on keyboards improve users' typing speed by enabling touch type. In this study, we investigated the feasibility of TypeBoard plus tactile landmarks. In summary, we evaluated users' typing experience on three settings: (1) regular software keyboard, (2) TypeBoard, and (3) TypeBoard with tactile landmarks.

5.1 Participants

We recruited 15 participants from the campus (aged from xx to xx, M = xx, SD = xx, xx females). All the participants were right handed. The have used software keyboards on smartphones for not less than xx years. xx of the 15 participants were familiar with tablet keyboards. Participants did not take part in the last two experiments.

5.2 Design and Procedure

The study followed a within-subject design to compare the typing experience in three keyboard setting (figure xx).

(1) **Config. 1):** *Regular Software Keyboard.* This setting represent the tablet keyboard we use in daily life. All contacts on the key are considered as keystrokes. Users need to hang their wrists in the air to avoid unintentional touches.
(2) **Config. 2):** *TypeBoard.* TypeBoard is a software keyboard with unintentional touches rejection. Only intended touches are considered as keystrokes. Users can rest their hand on the keyboard.
(3) **Config. 3):** *TypeBoard + Tactile Landmarks.* To afford tactile landmarks on TypeBoard, we attached 0.05mm thick stickers to the position of each key. There were small bumps on the F and J keys, which is the same as the physical keyboard. Users could position their hands without having to look at the keyboard.

The experiment contained three sessions of keyboard settings. In each session, participants filled in a document to complete five text entry tasks, including the four tasks we had introduced in the last two studies. In this study, we had an additional transcription task, where participants transcribed a Chinese phrase for five times as quick and accurate as possible. The transcription task is widely used in text entry studies [xx,xx,xx] to evaluate the ceiling typing speed. We counterbalanced both the order of keyboard settings and the order of tasks using a balanced latin square.

【图：实验三要对比的三种实验设置】

Participants had five minutes to warm up before they started each session. In the training phrase, participants entered words freely using the corresponding keyboard. Participants rested for five minutes between sessions to avoid fatigue. In average, participant spent xx minute to complete the experiment.

5.3 Result: TypeBoard vs. Regular Board

我们首先讨论有/无触觉反馈对用户打字可用性、效率的影响，纹理landmark的问题会在稍后讨论。

我们采用了什么样的统计分析方法，对于xx等违反正态分布的量，我们通过xx方法来进行校正。如果一个独立变量对结果有显著性影响，我们采用xx方法来检验变量两两之间的显著性。

对用户行为、用户体验、输入效率的影响。

要注意的点：1. 有无landmarks两种情况下，用户盲打的时间对比；每个按键点云标准差的对比；误触数量的对比；错误率。

5.3.1 *Completion Time.* English.

在誊写任务下可以对比打字速度。

5.3.2 *Error Rate (Text Entry).* CER

UER

5.3.3 *Detected Unintentional Touches Percentage.*

5.3.4 *Time Components.*

5.3.5 *Subjective Rating and Feedback.* English.

物理负担（疲劳程度），心理负担，主观输入速度，主观输入准确率（误触准确率，而非选词准确率）。

将手指休息在键盘上的频次over 任务，技术。

5.3.6 *Percentage of EyeFree time.*

5.4 Result: TypeBoard with/without tactile landmarks

在上面的分析中我们已经看到，自然的防误触算法对用户打字的可用性、效率和用户体验都有好处。接下来，我们来讨论在防误触的情况下，有/无触觉landmarks对文本输入可用性的影响。

对用户行为、用户体验、输入效率的影响。

5.5 Discussion

6 DISCUSSION

此外，这份工作还有两点关于人机交互工作的思考。第一，在用户意图推理的相关工作中，存在的一个普遍的问题，即用户行为和技术实现之间存在相互影响的作用，而大部分的先前工作都忽略了这一效应的存在[xx]。第二，在文本输入研究相关领域，誊写任务是默认的评测标准，然而，誊写任务与常见的文本输入相比缺失了手指休息和思考这两个成分，许多工作在这两个未被评测的步骤上存在明显的局限性[xx]，在这些工作中仅评测誊写任务是有失偏颇的。我们认为以上两点值得学术界的更多关注。

不同语言的问题。分析一下不同按键对防误触算法带来的挑战，并表面英文中只涉及到字母、空格、逗号句号的输入需求，不会受到很大的影响。

7 LIMITATION AND FUTURE WORK

7.1 Limitation

实验二中，每个用户使用TypeBoard的时长只有20分钟，也就是说，我们采集了用户在初次使用防误触键盘时的用户行为，但没有采集到长期影响下的用户行为。在未来，一个长期的实验是有必要和有价值的。

7.2 Future Work

有可能的follow up works：

键盘、触摸板一体for个人笔记本电脑？

触屏+纹理=触屏盲打？

利用手掌位置来处理触屏输入法中的漂移问题。

8 CONCLUSION

一些结论。

REFERENCES

[1] Michel Amberg, Frédéric Giraud, Betty Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. 2011. STIMTAC: a tactile input device with programmable friction. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*. 7–8.

[2] Michelle Annett, Fraser Anderson, Walter F Bischof, and Anoop Gupta. 2014. The pen is mightier: understanding stylus behaviour while inking on tablets. In *Proceedings of Graphics Interface 2014*. 193–200.

[3] Michelle Annett, Anoop Gupta, and Walter F Bischof. 2014. Exploring and understanding unintended touch during direct pen interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 5 (2014), 1–39.

[4] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 283–292.

[5] Jörg Brakensiek and Raja Bose. 2013. Method and apparatus for precluding operations associated with accidental touch inputs. US Patent App. 13/221,344.

[6] Olumuyiwa M Durojaiye and David Abzarian. 2016. Classification of touch input as being unintended or intended. US Patent 9,430,085.

[7] Wolfgang Fallot-Burghardt, Morten Fjeld, C Speirs, S Ziegenspeck, Helmut Krueger, and Thomas Läubli. 2006. Touch&Type: a novel pointing device for notebook computers. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*. 465–468.

[8] Gregg S Goyins and Mark F Resman. 2001. Palm pressure rejection method and apparatus for touchscreens. US Patent 6,246,395.

[9] Jason Tyler Griffin. 2013. Touch screen palm input rejection. US Patent App. 13/469,354.

[10] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–29.

[11] Terho Kaikuranta. 2006. Method for preventing unintended touch pad input due to accidental touching. US Patent 6,985,137.

[12] Sunjun Kim and Geehyuk Lee. 2016. Tapboard 2: Simple and effective touchpad-like interaction on a multi-touch surface keyboard. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5163–5168.

[13] Sunjun Kim, Jeongmin Son, Geehyuk Lee, Hwan Kim, and Woohun Lee. 2013. TapBoard: making a touch screen keyboard more touchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 553–562.

[14] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[15] Huy Viet Le, Sven Mayer, Benedict Steuerlein, and Niels Henze. 2019. Investigating Unintended Inputs for One-Handed Touch Interaction Beyond the Touchscreen. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–14.

[16] Vincent Levesque, Louise Oram, Karon MacLean, Andy Cockburn, Nicholas D Marchuk, Dan Johnson, J Edward Colgate, and Michael A Peshkin. 2011. Enhancing physicality in touch interaction with programmable friction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2481–2490.

[17] Hao Lu and Yang Li. 2015. Gesture on: Enabling always-on touch gestures for fast mobile access from the device standby mode. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3355–3364.

[18] Juha Matero and Ashley Colley. 2012. Identifying unintentional touches on handheld touch screen devices. In *Proceedings of the Designing Interactive Systems Conference*. 506–509.

[19] Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. 2017. Hand contact shape recognition for posture-based tabletop widgets and interaction. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 3–11.

[20] Valentin Popescu. 2015. Touch Screen with Unintended Input Prevention. US Patent App. 14/764,742.

[21] Luca Rigazio, David Kryze, Philippe Morin, and YUN Tiffany. 2013. System and method for differentiating between intended and unintended user input on a touchpad. US Patent 8,502,787.

[22] Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E Hudson, and Chris Harrison. 2014. Probabilistic palm rejection using spatiotemporal touch features and iterative classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009–2012.

[23] TouchFire. 2015. The screen-top keyboard for ipad. Website. https://www.kickstarter.com/projects/touchfire/touchfire-the-screen-top-keyboard-for-ipad.

[24] Kagan Tumer and Joydeep Ghosh. 1996. Estimating the Bayes error rate through classifier combining. In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 2. IEEE, 695–699.

[25] Malte Weiss, Roger Jennings, Julie Wagner, Ramsin Khoshabeh, James D Hollan, and Jan Borchers. 2008. Slap: Silicone illuminated active peripherals. *Ext. Abstracts of Tabletop* 8 (2008).

[26] Xuhai Xu, Chun Yu, Yuntao Wang, and Yuanchun Shi. 2020. Recognizing Unintentional Touch on Interactive Tabletop. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–24.

[27] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, et al. 2019. Sensing posture-aware pen+ touch interaction on tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.