TypeBoard: Identifying Unintentional Touch on Pressure-Sensitive Touchscreen Keyboards

ANONYMITY, address, Country

Text input is essential in tablet computer interaction. However, tablet software keyboards face the problem of misrecognizing unintentional touch, which affects efficiency and usability [29, 49]. In this paper, we proposed TypeBoard, a pressure-sensitive touchscreen keyboard that prevents unintentional touches. The TypeBoard allows users to rest their fingers on the touchscreen, which changes the user behavior. On average, users produce 40.83 unintentional touches every 100 keystrokes. The TypeBoard prevents unintentional touch with an accuracy of 98.88%. A typing study showed that the TypeBoard relieved fatigue ($p < 0.005$), reduced corrected error rate ($p < 0.01$), and improved the tablet keyboard' typing speed by 11.78% ($p < 0.005$). Under the premise that users could rest their fingers on the touchscreen, we added tactile landmarks on the TypeBoard, allowing users to find the keys through touch rather than sight. The tactile landmarks further improve the typing speed, outperforming the ordinary tablet keyboard by 21.19% ($p < 0.001$). The results show that pressure-sensitive touchscreen keyboards can prevent unintentional touch, which improves efficiency and usability from many aspects, including relieving fatigue, reducing error rates, and mediately allowing touch typing on the tablet.

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 Pressure-Sensitive Touchscreen Keyboards. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 25 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Increasingly more people are using tablets for text input tasks such as searching the Internet, writing email messages, and office work [53]. They use software keyboards on the touchscreens of the tablets. However, there is a gap between the tablet keyboard and the physical keyboard in aspects of the fatigue problem [27], switching of visual attention [10, 36, 47], and typing speed [17, 45]. Users can rest their fingers on the physical keyboard but cannot rest on the tablet keyboard because touching the screen causes misrecognition. The resting behavior plays an important role in the usability of physical keyboards. First, it reduces fatigue [29]. Second, users align their fingers by touching the keys and achieve touch typing [13, 19, 34, 37]. Touch typists type quickly because they do not use the sense of sight to find the keys. To bridge the gap between tablet keyboard and the physical keyboards, we proposed TypeBoard, a software keyboard that prevents unintentional touch. As figure 1 shows, the TypeBoard allows users to rest their fingers on the touchscreen. The TypeBoard prevents unintentional touches such as fingers resting and thenar eminence touches while passing users' typing behaviors. Furthermore, if we provide tactile landmarks on the TypeBoard through deformable screens [2], and changeable surface texture [5, 9, 33], the TypeBoard can support touch typing.

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

2476-1249/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

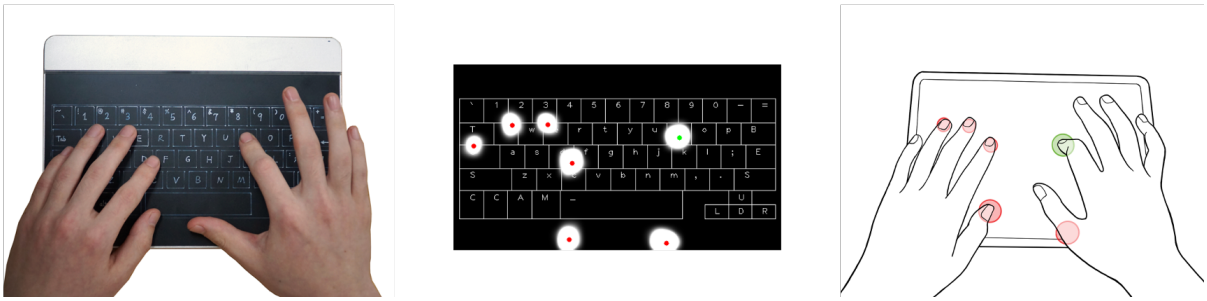


Fig. 1. The TypeBoard is a software keyboard that prevent unintentional touch such as fingers resting and thenar eminence touches.

We are not the first to explore the detection of unintentional touch on the touchscreen keyboard. In 2013, Kim et al. proposed TapBoard [29], a touchscreen keyboard that regards tapping actions as keystrokes and the other touches as unintentional touches. Tapping actions were those touches that neither exceed a timeout threshold of 450 ms nor exceed a displacement threshold of 15 mm. Users adapted to the thresholds, which led to unnatural typing behavior. TapBoard cannot judge the intention of touch until the touch is released. Even so, the accuracy is only 97%. We have two strategies to overcome TapBoard's limitations of naturalness and recognition performance. First, to ensure natural interaction, we defined *unintentional touches* as those touches that do not intend to input words. Second, we conducted user studies to understand user behavior, based on which we designed the algorithm of TypeBoard.

Techniques change user behavior. For example, users will not rest their fingers on the tablet keyboard unless the touchscreen prevents unintentional touches. Thus, to design TypeBoard, it is valuable to understand the users' typing behavior on the TypeBoard itself. However, how can we observe the user behavior on the TypeBoard before we have developed the TypeBoard? To solve this "chicken and egg" problem, we followed an iterative process, i.e., we developed a semi-finished TypeBoard, then analyze the user behavior on it, and finally improved the technique by using the latest dataset. In details, we conducted three user studies, each contributing to answering one of the three research questions as follows:

(1) **RQ1):** *What is the user behavior on an imaginary TypeBoard?* We conducted study one to collect users' typing data on a touchpad that has no feedback. Participants could not input words, instead, they pretended to input words, and imagined that the touchpad prevents unintentional touch. We found eleven categories of unintentional touches. The three most common ones were multiple finger resting, hypothenar eminence touches, and repeated touch reporting. We developed a semi-finished TypeBoard based on the analysis of user behavior.
(2) **RQ2):** *What is the user behavior on the TypeBoard?* We conducted study two to collect users' typing data on the (semi-finished) TypeBoard. We did not found new unintentional touch behavior. However, the user behavior in study two was different from study one in details. For example, the average force of intentional touch was 33.78% lighter in study two because users learned that they could type without much effort, gradually making the touch lighter. We used the data in study two to improve the TypeBoard, achieving an accuracy of 98.88%, with a delay of 100 ms.
(3) **RQ3):** *What is the performance of the TypeBoard? What if there are tactile landmarks on the TypeBoard?* We conducted study three to evaluate users' typing performance on three settings, including ordinary tablet keyboard, TypeBoard, and TypeBoard plus (the TypeBoard with tactile landmarks). The result shows that the TypeBoard improves the efficiency and usability of touchscreen typing from many aspects, including

avoiding fatigue, relieving cognitive load, and reducing error rates. The TypeBoard plus further improves the performance of the TypeBoard by allowing touch typing on the tablet.

The contribution of this work is three-fold. First, we proposed TypeBoard, which identifies unintentional touch in text input tasks with high accuracy (98.88%) and low latency (100 ms). Second, we explored the user behavior on the touchscreen keyboard with unintentional touch prevention. We published the dataset. Third, we empirically demonstrated the advantages of the TypeBoard in typing speed, error rate and subjective user experience.

2 RELATED WORK

2.1 Unintentional Touch Prevention on Touchscreen

Touch is the main input channel on touchscreen devices such as smartphones, tablets, and tabletops. 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 [52], or namely, accidental/unwanted touches [35, 40]. If the unintentional touch triggers a response, it will affect the interaction's efficiency, and naturalness [6, 52]. Unintentional touch will trigger unwanted responses and disrupt the user's workflow. Then, the user takes extra time to cancel the accidentally triggered operation. As time passes, the user tends to behave carefully and prudently on the device to avoid triggering unwanted touch events. For example, users of ordinary touchscreen keyboards hover their fingers over the screen to avoid accidental palm touches, which leads to the fatigue problem [49].

However, unintentional touch is inevitable in touchscreen interactions. For example, the thenar eminence on the human hand will constantly contact the touchscreen during the daily use of smartphones [31]. Fortunately, we can filter out unintentional touches by software techniques. In the literature, methods of preventing unintentional touches have been extensively studied. We introduce related work through two taxonomies: (1) use scenarios and (2) sensors.

2.1.1 Unintentional Touch Prevention over use scenarios. The definition of unintentional touch varies in different use scenarios. When the scenario is not limited, unintentional touches refer to those touches that do not contribute to any interaction goal [52]. It is more clear to determine the intentions of touches in specific tasks. For example, unintentional touches in text entry tasks are those that do not intend to input words.

A few techniques [40, 52] and a large amount of patents [11, 15, 26, 42, 43] identify unintentional touch in any scenario. Metero et al. presented guidelines to reduce the number of unintentional touches on smartphones [40]. Their filtering criteria rejected 79.6% of unintentional touches. Xu et al. identify and filter out unintentional touches on the interactive tabletop using gaze direction, head orientation, and screen contact data [52]. The accuracy was 91.3%. These approaches suffered from a low recognition rate.

In the literature, more studies investigated unintentional touch in specific scenarios. TapBoard and TapBoard2 discussed this issue under the text entry task. TapBoard recognized tapping actions as typing behaviors, while other touches were unintentional touches. Tapping actions were those the duration is shorter than 450 ms, and the movement is within 15 mm. Users needed to adapt to the thresholds, which is unnatural. Even so, the recognition rate of TapBoard was not high (97%). TapBoard2 was an improved version that distinguished between typing and pointing actions with an accuracy of 95%. In pen and tablet interaction, unintentional touch is one of the most prominent features identified as problematic by users [6]. For example, users had to write in an uncomfortable position to avoid palm touch. Several studies investigated the prevention of unintentional touch in pen and tablet interaction [7, 22, 44]. Schwarz et al. used spatiotemporal touch features to train the classifier [44] that achieved the best self-reported performance, reducing accidental palm touch to 0.016 per pen stroke,

while correctly passing 98% of stylus inputs. In general, methods of unintentional touch prevention perform better when the scenario is limited.

2.1.2 Unintentional Touch Prevention over sensors. A mass of studies have been conducted to recognize unintentional input on touchscreen devices, including smartphones [31, 32, 35, 40], tablets [7, 26, 28, 29, 44] and tabletops [52]. Most techniques only leverage built-in touchscreen signals, including touchpoint information [26, 28, 29, 40, 44] and capacitive images [7, 31]. Metero et al. used touchpoint patterns to prevent unintentional touch on smartphones [40]. They proposed filtering criteria such as touch duration, position, and trajectory pattern that rejected 79.6% of unintentional touches while rejecting 0.8% of intentional touches. chwarz et al. presented a filter that distinguishes between legitimate stylus and palm touches on tablet computers [44]. They extracted features from touchpoints and used the decision forest to train a machine learning model, which reduced accidental palm inputs to 0.016 per pen stroke, and correctly passing 97.9% of stylus inputs. PalmTouch [31] is a smartphone technique that distinguishes between palm touch and finger touch. PalmTouch leveraged the touchscreen's capacitive image as input and used Convolutional Neural Network (CNN) as the method, resulting in an accuracy of 99.53% in realistic scenarios.

Other techniques enhanced the sensing ability to improve the performance [21, 32, 35, 52]. GestureOn [35] distinguishes intended gesture input from unintentional touches in the standby mode of smartphones. The user can trigger gesture shortcuts before turning on the screen. GestureOn used most smartphone sensors, including proximity sensors, light sensors, IR sensors, and Inertial Measurement Unit (IMU). The system also leveraged the pressure signal on the touchscreen, which is not popularized on smartphones yet. Based on sensor fusion, GestureOn acquired 98.2% precision and 97.6% recall. Xu et al. leveraged gaze direction, head orientation, and screen contact data to identify unintentional touches on interactive tabletop [52]. The result showed that the patterns of gaze direction and head orientation improved the accuracy by 4.3%, reaching 91.3%. In general, sensor fusion unsurprisingly improved the recognition rate of unintentional touch.

2.1.3 Summary. We suggest two viewpoints. First, in general, methods of preventing unintentional touch perform better when the use scenario is limited. In specific scenarios, we have the opportunity to prevent unintentional touches with high accuracy so that users are willing to change their behavior to acquire benefits from the techniques [6, 28, 29]. Second, sensor fusion can usually improve the recognition rate of unintentional touch. For example, several studies have proved that additional input channels such as pressure signals [35], gaze direction, and head orientation [52] provide help in recognition of unintentional touch. In this paper, we investigate the identification of unintentional touch on the pressure-sensitive touchscreen keyboard. Because we used rich sensor signals to detect unintentional touches in a specific task (text input), we expected a high recognition rate. In this situation, we were interested in a more profound question: can our proposal change the user behavior, be more natural and relaxing?

2.2 Benefits from Unintentional Touch Prevention

A direct benefit from unintentional touch prevention is to reduce the harm caused by the misrecognition, which improves efficiency and interaction naturalness. Besides this, we summarize other advantages of unintentional touch prevention as follows.

2.2.1 Bridge the Gap Between Software Keyboards and Physical Keyboards. More and more people use software keyboards on the touchscreens of the tablets [53]. However, software keyboards cannot compare to physical keyboards in usability [10, 27, 36, 47] and efficiency [12, 17, 45]. On physical keyboards, users can rest their fingers on the buttons, which is a crucial usability factor of physical keyboards. First, it reduces fatigue [29]. Second, users can input quicker through touch typing, using the sense of touch (instead of sight) to find the keys [13, 19, 34, 37]. Since the prevention of unintentional touch allows tablet users to rest their fingers on

touchscreens, it reduces typing fatigue. Furthermore, on the premise that the user can touch the screen without triggering responses, we can provide tactile landmarks through deformable screens [2] or changeable surface texture [5, 9, 33], so that the users can perform touch typing. In this way, unintentional touch prevention bridges the gap between software keyboards and physical keyboards.

2.2.2 Leveraging Unintentional Touch as Input Channels. The goal of our work is to filter out unintentional touches on the touchscreen. However, some literature tried to use unintentional touches for situational awareness [55] and explicit interactions [28, 31, 41]. Zhang et al. leveraged unintentional touches such as the palm touch to augment pen and touch interactions, e.g., providing "Palm Tools" that teleports to a hand-centric location when the user plants their palm on the screen while drawing [55]. Koura et al. [30] proposed to use the forearm, which often creates problems such as incorrect recognition and occlusions, as a new input channel to manipulate menu and data storage. Matulic et al. enriched tabletop interaction by considering the whole hand as input [41]. The system detects seven different contact shapes with 91% accuracy and can be used to trigger, parameterize, and dynamically control menu and tool widgets. PalmTouch is an additional input modality that distinguishes between finger touches and palm touches [31]. PalmTouch can be used as a shortcut to improve reachability. TapBoard2 [28] distinguishes between typing and pointing actions, thereby unifying the keyboard and mouse control spaces. It depresses the burden of frequently switching between devices [16].

3 STUDY 1: USER BEHAVIOR ON AN IMAGINARY TYPEBOARD

In this study, we collected data from participants' typing actions on a pressure-sensitive touchpad. The motivation was to investigate users' typing behavior on an imaginary touchscreen keyboard that prevents unintentional touch, based on which we could design the algorithm of unintentional touch prevention. Because (incorrect) feedback would affect users' behaviors, participants typed on a touchpad without any feedback in this study. Participants could not input words by typing. Instead, they imagined that the desired words are inputted. Besides, participants needed to adapt their behaviors according to the imagination that the keyboard can prevent unintentional touch.

3.1 Participants

We recruited 16 participants from the campus (aged from 19 to 26, M = 22.13, SD = 2.13, eight females). All the participants were right-handed and native Chinese speakers. They have used software keyboards on smartphones for not less than two years (M=7.50, SD = 2.25). Eight participants have ever used software keyboards on tablets.

3.2 Design and Procedure

Figure 2 illustrates the experimental setting. There was a Sensel Morph [4] pressure-sensitive touchpad on the desk. We placed a Windows Surface tablet computer on the touchpad, covering half of the touchpad. We drew a QWERTY layout on the touchpad using highlighter pens. The devices were a substitute for the pressure-sensitive touchscreen, which is not available in markets yet. Participants filled in a Microsoft Word document to complete the experimental tasks. They touched on the tablet to select table cell and typed on the touchpad to imagine inputting words. The system recorded every touch and the screencast during the experiment.

The experiment included four text input tasks: (1) filling in personal information, (2) short questions, (3) open-book examination, and (4) picture writing. The tasks were in Chinese. We counterbalanced the order of tasks using a balanced latin square. We did not include transcription as a task as many other text entry studies do [38, 46, 54], 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 detail of tasks is as follows:

(1) **Filling in personal information:** There was a table of ten blanks about personal information, such as name and gender. Figure 3(a) shows the examples in Chinese and the corresponding translation. This assignment represented those tasks that require frequent switching between text input and cursor control.

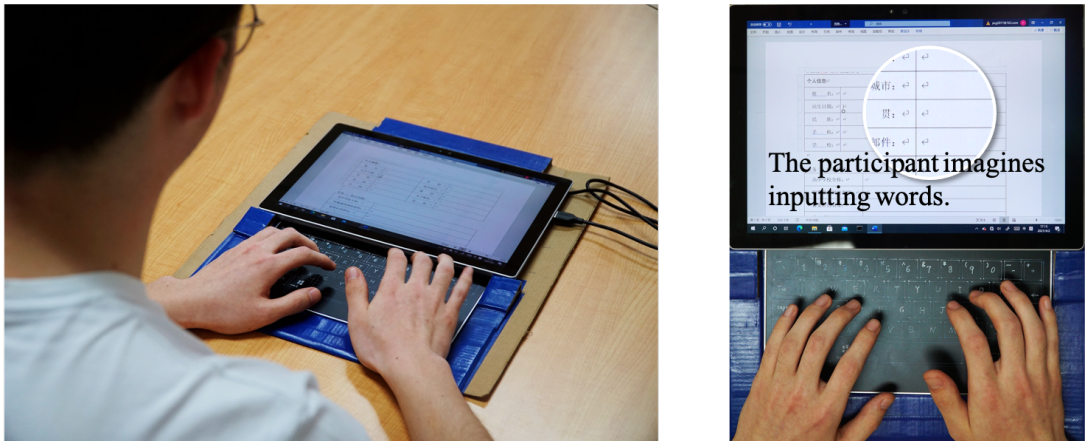


Fig. 2. The experimental setting of study one. We placed a tablet and a pressure-sensitive touchpad together as a substitute for the pressure-sensitive touchscreen. The keyboard had no feedback. The participant imagined inputting words.

To protect privacy, participants felt free to fill in fake information. However, participants should remember what they intended to input, which is crucial for the subsequent process of labeling data.

(2) **Short questions:** There was a table of ten short questions, such as "the favorite color" and "the best friend". Participants were allowed to fill in a fake answer.

(3) **Open-book examination:** The exam consisted of five hard questions, such as "what is the 50th element on the periodic table?". Participants could hardly know the answers, so they needed to search the Internet. This assignment represented the common task of browsing websites. Because the participants could not input words in the search engine, they said as they wrote, so the experimenter could replace them to input the words.

(4) **Picture writing:** As figure 3(b) shows, participants described the picture in five sentences. This assignment represented the tasks of writing articles. Participants said as they wrote, so the experimenter could replace them to input the words.

(a)

个人信息			
姓　　名：		性　　别：	
出生日期：		现居城市：	

Personal information			
Name：		Gender：	
Birthday：		Address：	

(b)

Fig. 3. The illustration of experimental tasks. The left side shows the examples of task one (filling in personal information) in both Chinese and translation. The right side is the figure we used in task four (picture writing).

Before the experiment, the participant had five minutes to get familiar with the experimental requirement. In the warm-up phase, participants typed on the touchpad freely. We reminded the participants of two points. First, the keyboard did not provide any feedback. Participants could not input words but imagined inputting words. As

the Chinese text entry method involved word selection, participants assumed that the desired word is always the first in the candidate list. Second, users needed to imagine that the keyboard prevents unintentional touches and adjust their behavior according to this assumption. For example, they could rest their fingers on the keyboard while thinking.

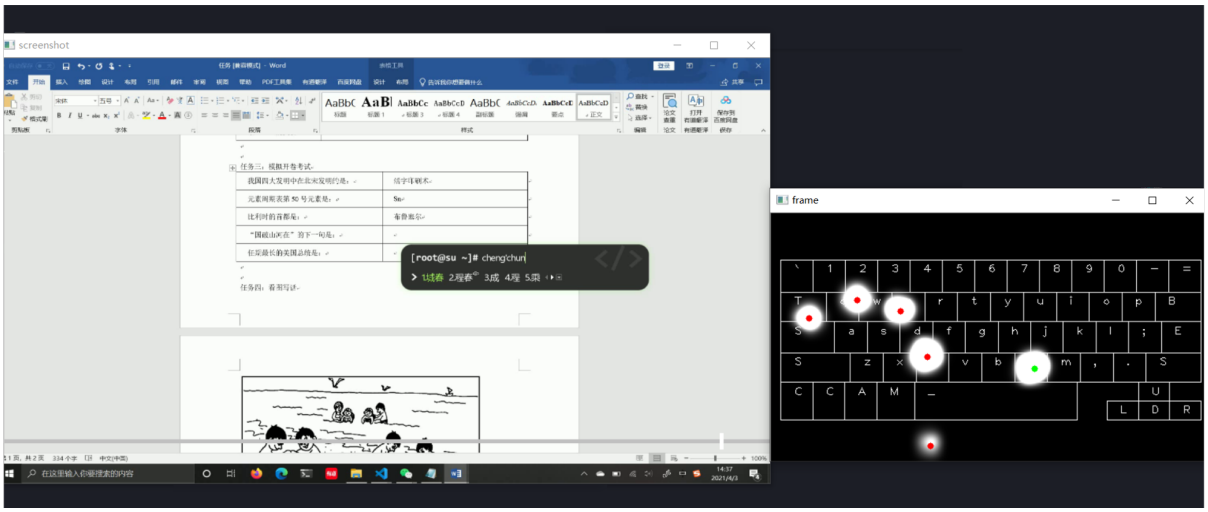


Fig. 4. The illustration of the label program.

After each task, the participant labeled the data through an interactive program as figure 4 shows. The program showed the touchpad capacitive images and the tablet screencast at the same time. There were red points on capacitive images that showed the touchpoints. Participants labeled the intended touches as green points. Participants were able to judge most touches because they could get context from the screencast. If participants were not sure, they labeled the touchpoints as blue points to remove the data. On average, participants spent five minutes finishing the text input tasks and spent 45 minutes labeling the data. Participants rested for five minutes between two tasks to avoid fatigue. The study lasted for 70 minutes.

3.3 Apparatus

As figure 2(right) shows, we placed a Windows Surface tablet computer and a Sensel Morph pressure-sensitive touchpad [4] together as a substitute for the pressure-sensitive touchscreen. The Sensel Morph senses 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, ranging from 5g to 5kg. The maximal frequency of the Sensel Morph was 125Hz (8ms latency). We slowed the frequency 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 device is so sensitive that almost every touch is recognized. Thus, in this paper, we identified unintentional touches among reported touchpoints while ignoring the Sensel Morph's missing touches.

The size of the sensing area on the Sensel Morph was 240 mm x 138 mm. We used highlighter pens to draw a Qwerty layout on the touchpad. The Sensel Morph width (240mm) was shorter than the keyboard on a 15 inches MacBook (270mm). We removed some keys that are less frequently used, such as square brackets and the semicolon, so that the Qwerty layout could be placed in the Sensel Morph, while each key's size remained the same as Macbook. Because the Qwerty layout is changeable in the software keyboard, we did not leverage the

layout as prior knowledge to recognize unintentional touch. The tablet computer was Windows Surface Pro6, with i7 Intel Core Processor. The program ran on the tablet at 50 FPS.

3.4 Result

The dataset contains 12659 touches, excluding the ambiguous touches (0.18%) in the labeling process. 67.5% of the data were positive samples (intentional touches), while 32.5% were negative samples (unintentional touches). Based on the data, we developed the TypeBoard in three steps, each contributing to solving one of the critical problems as follows:

(1) **Model V1):** *data processing*. There were some mislabeled data points because some participants misunderstood the concept of unintentional touches. We developed a naive Support Vector Machine (SVM) model to identify unintentional touch. If there was a vast difference between the predicting results and the labels, we asked the participants to relabel the suspicious data points through e-mails.
(2) **Model V2):** *filtering multiple fingers resting*. Observation suggested that multiple fingers resting was the most frequent unintentional touch, where users rested more than two fingers on the screen. We added targeted criteria into the SVM model to filter out multiple fingers resting.
(3) **Model V3):** *understanding the user behavior*. We analyzed the fail cases of Model **V2** to understand those user behaviors that challenged the algorithm, based on which we further improved the SVM model.

Table 1 shows the feature vector we used to train the SVM model in each step and the recognition results. In the remainder of this subsection, we introduced the unintentional touch identification model in detail.

3.4.1 Model V1: naive model for data processing. In the first step, we trained a naive SVM model to identify unintentional touch using straightforward features. We sampled the first five frames (100 ms) of each touch. If the touch duration is shorter than five frames, we sampled the whole touch. As table 1 (**V1**) shows, we extracted features from the samples as follows: for the contact area, ellipticity, displacement, force, and intensity over frames, we calculated the temporal features, 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, namely Model **V1**. We balanced positive and negative samples in weight.

We used the model to simulate the dataset. We found that some participants misunderstood the concept of unintentional touches, for example, regarding incorrect character inputs as unintentional touches. We asked the participants to relabel the suspicious data points through e-mail. After the label correction, we had 12624 data points, including 68.38% positive samples and 31.62% negative samples. Leave one out cross-validation shows that the accuracy of Model **V1** was 96.86% (SD=4.17%).

3.4.2 Model V2: filtering multiple fingers resting. Observation showed that most unintentional touches (72.66%) were caused by multiple fingers resting, where users rested no less than three fingers on the screen simultaneously. The resting fingers contact the screen one after another. After the first touch, the following touches come within 100 ms in most instances. In 86.05% cases, the second finger touches within 100 ms, while in 76.30% cases, the third finger also touches within 100 ms. Because our model has a latency of 100 ms, the model has a big chance to realize that multiple fingers are resting. Thus, we could design targeted features to filter out the unintentional touches caused by multiple fingers resting. We added a series of features in Model **V2** to deal with this problem. The criterion was the proportion of the touch's pressure, intensity, and contact area among all touches. Table 1 (**V2**) shows the details. Leave one out cross-validation shows that Model **V2** increased the recognition accuracy to 98.59%.

3.4.3 Model V3: understanding the user behavior to improve the model. We analyzed the fail cases of Model **V2** to understand those user behaviors that challenged the model. The error rate of Model **V2** was 1.41%, including 1.00% false positives and 0.41% false negatives. The false positives referred to the misrecognized unintentional

Table 1. The features we fed into the SVM model and the accuracy among model versions. For the features except "the relationship to recent/nearby touches", we extracted the temporal features over frames, including maximum, minimum, mean, skewness, and kurtosis.

Version	Criterion	Feature	Introduction	Accuracy
V1	The information of current touch	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.	
		Pressure	The contact force in grams.	
		Intensity	The ratio of the pressure to the contact area.	
V2	The proportion	Pressure proportion	The ratio of the pressure to the total pressure of all touches.	98.59% (SD=1.46%)
		Intensity proportion	The ratio of the intensity to the total intensity of all touches.	
		Area proportion	The ratio of the contact area to the total contact area of all touches.	
V3	The location of 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.	
	The 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.	

touches, while the false negatives were the missing intentional touches. As table 2 shows, we classified the fail cases into 16 categories. We counted each kind of fail case, discussed the reasons, and gave possible solutions. For the fail cases with a white background in the table, humans (the researchers) could judge their intentions without extracting contextual information from the experiment screencast. That is, the machine should have correctly predicted if it is as smart as a human. For these cases, we proposed features to improve the model. Here are some examples.

(1) **EG1):** *Hypothenar eminence.* As figure 5(a) shows, the hypothenar eminence refers to a group of muscles of the palm that control the motion of the little finger, while the thenar eminence is the group of muscles on the palm at the base of the thumb. Both the two eminences may contact the touchscreen when a user is typing. In particular, the touches caused by the hypothenar eminence are usually heavy and intensive, which is easy to confuse with intentional touches. Fortunately, these touches are in the bottom left and the bottom right corners. Thus, the distance to the closest corner could be a powerful feature to reject these unintentional touches.
(2) **EG2):** *Continuous touches.* When a user continuously typed on the same key (e.g., the backspace), the following touches were lighter than the first touch ($p < 0.005$). Among the continuous touches, the average pressure of the first touch was 167.01 g (SD = 106.76), while the average pressure of the following touches was 150.42 g (SD = 112.70). Because the following touches are light, they are likely to be recognized as unintentional touches. Information of recent touches help to correct this fail case.
(3) **EG3):** *One-hand typing.* As figure 5(b) shows, sometimes the participant typed with one hand while resting the other hand on the touchscreen. In this situation, the Model **V2** mistakenly believed that all the touches were unintentional touches caused by the multiple fingers resting behavior. We added information of nearby touches to correct this fail case because the typing finger is usually far away from the resting fingers.

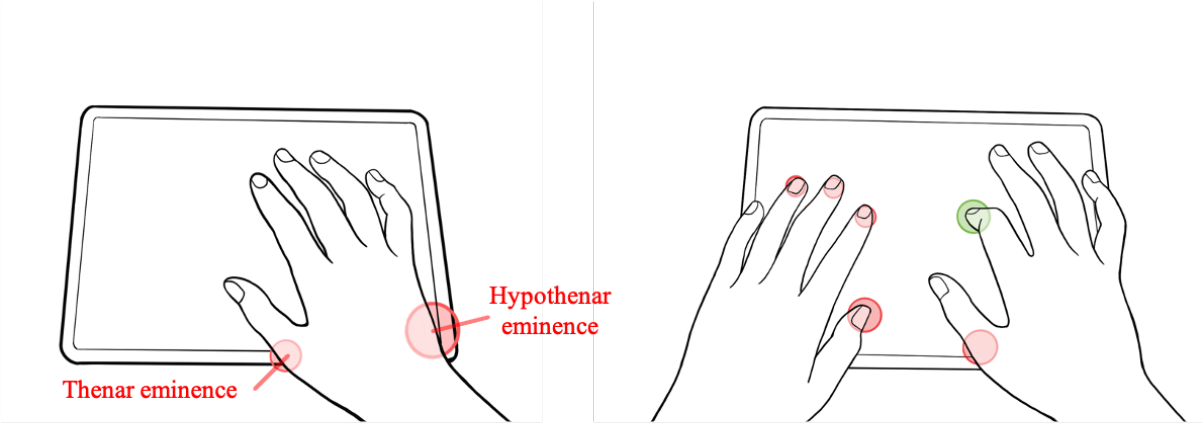


Fig. 5. The examples of fail cases. The fail cases reveal those user behaviors that challenged the algorithm.

However, for the fail cases with gray background in table 2, humans (the researchers) could judge their intentions without watching the experiment screencast. We deemed that these fail cases are inevitable because the machine can not know what the user will input in advance. Here are some examples. First, sometimes the participant rested one finger on the touchscreen heavily, indistinguishable from an intentional touch. Second, the participant performed a very light touch during inputting a word, which is indistinguishable from an unintentional touch. Thus, the model's accuracy has a certain upper limit (roughly 99.40% in this dataset), while our goal is to approach it.

According to the user behaviors summarized in table 2, we added two criteria in the model training. The first one is the location of the touch, including the minimum distances to the edges and the corners. We did not leverage the prior knowledge of the keyboard layout because the software keyboard layout is changeable, while we wanted a universal model. The second criterion is the relationships between the current touch and the

Table 2. The fail cases of Model **V2**. The "N" column refers to the counting of each case.

Cases	Introduction	N	Helpful features?
False Positives			
Hypothenar eminence (figure 5a)	The hypothenar eminence usually contacts the screen while typing.	23	Touchpoint location, e.g., nearing the corners?
Thenar eminence (figure 5a)	The thenar eminence usually contacts the screen while typing.	2	Touchpoint location, e.g., nearing the bottom edge?
Repeated reporting (spatial)	A touch is misrecognized as two touches when the contact area is large.	12	Info. of nearby touchpoints.
Repeated reporing (temporal)	A touch is misrecognized as two touches if the touch is nearly released midway.	3	Info. of recent touchpoints.
Edge touch	Users trigger unintentional edge touch when adjusting the placement of devices.	7	Touchpoint location, e.g., nearing the flanks?
Two fingers resting	The user rests two fingers on the screen.	3	Info. of nearby touchpoints.
Extra touchpoint (light)	When inputting, users trigger extra touchpoints, which are lighter than the recent intentional touches.	9	Is this touch lighter than the recent touches?
Slide	A slide is less likely to be an intentional keystroke.	7	Touchpoint displacement.
One finger resting	Users rest one finger on the touchsreen, which is indistinguishable from an intentional touch.	9	No solution.
Extra touchpoint (heavy)	When inputting, users trigger extra touches heavily, which seems like an intentional touch.	15	No solution.
False Negatives			
Continuous touches	When a user continuously type on the same key (e.g., the delete key), the following touches will be lighter.	13	Info. of recent touchpoints.
Rollover-typing	The next key is pressed before the previous is released [14].	12	info. of recent touchpoints.
One-hand typing	The user types with one hand, while the other hand is resting on the screen. This case is similar to multiple fingers resting.	6	info. of nearby touchpoints.
Palm touch	The use types a key when the palm is resting on the screen. If the palm touch 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 is indistinguishable from an unintentional touch.	46	No solution.
Small contact area	The intended touch with a very small contact area. This seems like an unintentional touch.	6	No solution.

recent/nearby touches. The features in detail are in table 1 (**V3**). The Model **V3** used all the features in the table. 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 0.56% false positives and 0.37% false negatives. So far, we have obtained a model with high recognition accuracy. However, as we said in the introduction, the data collected in study one did not represent the user behavior on the software keyboard with unintentional touch prevention, so we named Model **V3** as semi-finished TypeBoard. In study two, we collected the user behavior on the semi-finished TypeBoard and then used the new data to improve the model.

4 STUDY 2: USER BEHAVIOR ON THE TYPEBOARD

In this study, we obtained users' typing data on the semi-finished TypeBoard, the touchscreen keyboard developed in study one that prevents unintentional touch. The motivation was to investigate users' typing behavior, based on which we can further improve the TypeBoard.

4.1 Participants

We recruited 16 participants from the local campus (aged from 19 to 37, M = 22.19, SD = 4.55, six females). All the participants were right-handed and did not take part in the first study. They have used software keyboards on smartphones for not less than one year (M = 5.88, SD = 2.36). Ten participants have ever used software keyboards on tablets.

4.2 Design and Procedure

As figure 6 shows, the experimental devices were the same as those in study one, including a Windows Surface tablet computer and a Sensel Morph pressure-sensitive touchpad. The working frequency was 50 FPS. There were text entry tasks, which were the same as study one, namely filling in personal information, short questions, open-book examination, and picture writing. The differences in study two are as follows. First, participants could input words. Second, the keyboard prevented unintentional touch. Intentional touches would trigger keystrokes and audio feedback, while the system ignored unintentional touches.

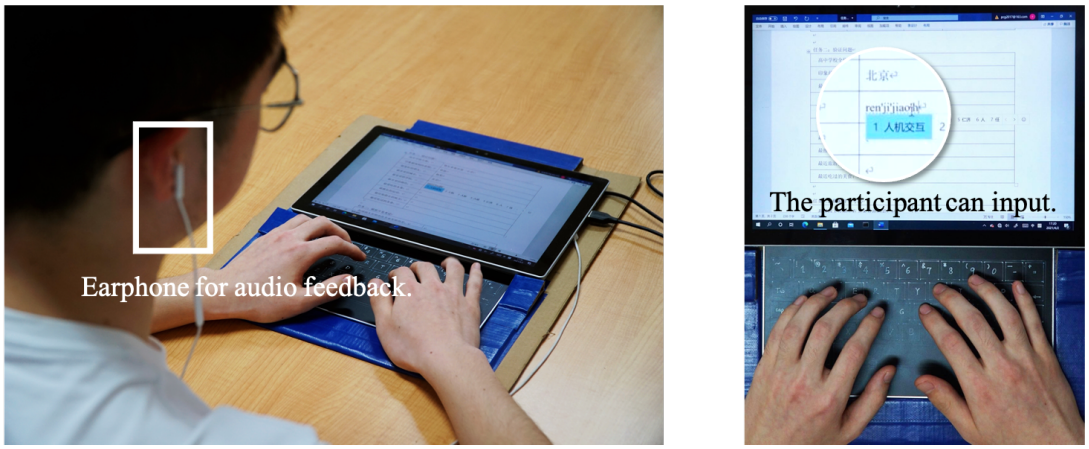


Fig. 6. The experimental setting of study two.

Before the experiment, participants warmed up through five-minute free writing. Because no participant had experience typing on a keyboard with unintentional touch prevention, we reminded the participants that they

could rest their fingers on the keyboard. Participants could decide to rest their fingers or not according to the task and their preference. After finishing each task, the participant labeled the data using the label program introduced in study one. During the labeling process, we compared participants' labels with the predictions by the semi-finished TypeBoard. When the two results were different, we recorded the data point for later processing. If the experimenter could not explain the difference, he would discuss it with the participant. On average, participants spent 10 minutes completing the text entry tasks and spent one hour labeling the data. Participants rested for five minutes between two tasks to avoid fatigue. The study lasted for 90 minutes. The experiment took more time than study one because participants needed to input correct words in this study.

4.3 Result

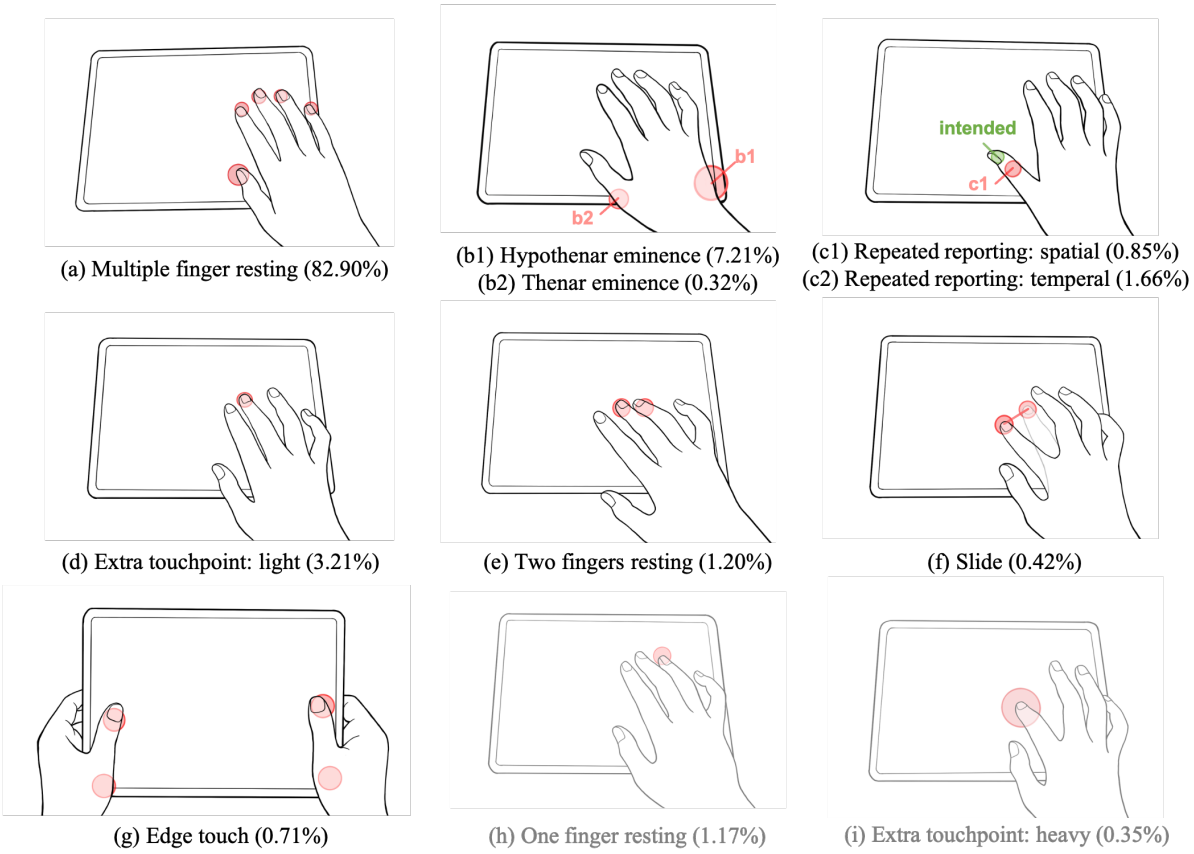
The dataset contained 13789 touches, excluding the ambiguous touches (0.22%) in the labeling process. After the double-check of labels, the dataset consisted of 71.01% positive samples (intentional touches) and 28.99% negative samples (unintentional touches). The semi-finished TypeBoard predicted the intention of touch with an accuracy of 98.05% (SD=1.51%) in study two. There were 0.39% (SD=0.37%) false positives and 1.56% (SD=1.37) false negatives. That is, participants encountered 2.20 unrecognized touchpoints and 0.55 false triggering touchpoints every 100 keystrokes in study two. The recognition rate of the semi-finished TypeBoard dropped from 99.07% on the study one dataset to 98.05% on the study two dataset. It indicates that the user behavior changed in study two, so the model trained on the study one dataset did not work perfectly in study two. Because the user behavior observed in study two was closer to the user's real usage, it is valuable to investigate the differences in user behaviors between the two studies.

Compared with study one, we did not find new cases of unintentional touches in this study. However, the user behavior in study two was different from the last study in details. Table 3 shows the examples. First, the user's average pressure of intentional touches in this study was significantly lighter than the last study ($t_{15} = 2.78, p < .01$). When typing with feedback, users found that they could type letters without much effort, gradually making the touch lighter. Second, there were more continuous touches in this study ($t_{15} = 3.57, p < .005$), where a touch is close to the last touch in both time intervals (< 500 ms) and distance (< 0.5 key widths). This is because participants need to remove incorrect words by continuously pressing the backspace in real typing tasks. Third, the study two dataset contained fewer rollover-typing than the study one ($t_{15} = -6.32, p < .001$). Rollover-typing means the next key is pressed before the previous is released. Participants typed slower in study two because they needed to enter correct words. This slower typing speed correlated with fewer keystrokes typed with rollover [14].

Table 3. The differences of user behavior between the two studies. We use t test to evaluate the significance of the difference. If Levene's test rejects the homoscedasticity of data, we use unequal variances t test instead.

Measure	**Introduction**	**Study one**	**Study two**	**Levene's test**	**T test**
Touch pressure	The average touch pressure of intentional touches in grams.	188.39g (SD=64.72)	124.75g (SD=60.71)	Reject	$t_{15} = 2.78$, $p < .01$
Continous touches	The continous touches as a percentage of all intentional touches.	4.02% (SD=1.96%)	11.89% (SD=4.41%)	Pass	$t_{15} = 3.57$, $p < .005$
Rollover-Typing	The rollover-typing touches as a percentage of all intentional touches.	17.60% (SD=9.34%)	7.73% (SD=5.22%)	Pass	$t_{15} = -6.32$, $p < .001$

As figure 7 shows, we counted the unintentional touches caused by each case of user behavior. The number was counted in touches, e.g., we counted the five fingers resting behavior five times. The three most common unintentional touches were multiple fingers resting (82.90%), hypothenar eminence touching (7.53%), and extra light touchpoint (3.21%). For the unintentional touches illustrated with translucent sub-figures, including (h) one finger resting and (i) extra heavy touchpoint, humans could not identify their intention without contextual information. The machine could hardly judge these cases. Fortunately, their proportion in all unintentional touches was only 1.52%. We retrained the model using the study two dataset. Leave one out cross-validation shows that the accuracy increased to 98.88% (SD=0.73%), significantly surpassed the semi-finish TypeBoard ($F = xx$, $p < xx$). There were 0.66% (SD=0.58%) false positives and 0.46% (SD=0.45%) false negatives. On average, TypeBoard users will encounter 0.65 unrecognized touchpoints and 0.93 false triggering touchpoints every 100 keystrokes. So far, we have finished the design of the TypeBoard.



(a) Multiple finger resting (82.90%)
(b1) Hypothenar eminence (7.21%)
(b2) Thenar eminence (0.32%)
(c1) Repeated reporting: spatial (0.85%)
(c2) Repeated reporting: temperal (1.66%)
(d) Extra touchpoint: light (3.21%)
(e) Two fingers resting (1.20%)
(f) Slide (0.42%)
(g) Edge touch (0.71%)
(h) One finger resting (1.17%)
(i) Extra touchpoint: heavy (0.35%)

Fig. 7. The classification of unintentional touches. The percentage in the bracket is the proportion of the corresponding cases in all unintentional touches.

4.4 Discussion

4.4.1 Comparison to previous work. TapBoard [29] was the latest study that similar to our work. Please note that the TapBoard is different from our proposal TypeBoard. We both investigate unintentional touch prevention on touchscreen keyboards. However, the TapBoard has two defects. First, the TapBoard recognizes tapping actions as keystrokes and the others as unintentional touches. Tapping actions were those the duration is shorter than 450 ms, and the displacement is within 15 ms. Users adapt their behavior to meet the technique, which is not natural and relaxing. Second, the TapBoard cannot judge a touch until the touch is released. Even so, the recognition rate is only 97%.

We evaluated the TapBoard on our study two dataset, where participants performed natural typing behaviors on the touchscreen keyboard that prevents unintentional touch. The recognition accuracy of the TapBoard was xx.x% on our dataset, which is nowhere near the performance of our proposal (98.88%). The result shows that our proposal outperforms existing work.

4.4.2 The user differences. As figure 8(left) shows, the user behaviors of different participants vary a lot. For extreme examples, P1 generated 74.40 unintentional touches every 100 keystrokes, while P16 only produced three unintentional touches during the experiment. We divided participants into three categories. The first type of users (P1-P10) were willing to rest their fingers on the touchscreen. They generated a lot of unintentional touches during the experiment. The second type of users (P11-P12) believed that TypeBoard could prevent unintentional touches. They put their palms on the touchpad but seldom rested their fingers on the keyboard. P12's comment was the key: "*I do not rest my fingers on the touchscreen because I cannot press down directly in the touched state.*" The third type of users (P13-P16) hanged the wrist to input. Surprisingly, they varied a lot in touchscreen keyboard expertise. P13 and P14 had only used touchscreen keyboards for one year, P15 had no experience, while P16 was an expert who performed touch typing on touchscreen keyboards. Results show that personalization is promising to improve the TypeBoard.

4.4.3 The variety of tasks. As figure 8(right) shows, the task had a significant effect on the number of unintentional touches (F,p). In the four experimental tasks, users produced xx.x (SD=xx.x), xx.x (SD=xx.x), xx.x (SD=xx.x), and xx.x (SD=xx.x) unintentional touches every 100 keystrokes. Bonferroni-corrected post hoc tests showed significant differences between the following task pairs: 1-3 (p<), 1-4 (p<), 2-3 (p<), 2-4 (p<). Users generated more unintentional touches in the first two tasks, which involved more frequent switching between text input and cursor control. Results show that the variety of tasks is important for exploring unintentional touch but usually being neglected in studies [29, 44, 52].

4.5 Why sample five frames in each touch?

The more frames we sample in each touch, the more accurate the prediction is. However, a long sampling window means a large delay (20 ms per frame), which affects the user experience. There is a trade-off between the delay and the recognition accuracy. To strike a balance, we simulated our algorithm with different delays. Figure 9(left) shows the results. An RM ANOVA showed that delay had a significant effect on recognition accuracy ($F_{4,56} = 4.88, p < 0.05$). Pair-wise comparisons showed significant differences between the following delay pairs: 60-100 (p<.05), 60-120 (p<.05), 60-140 (p<.05), 80-140 (p<.05). Thus, sampling five frames (100 ms) was the best choice, which resulted in acceptable prediction accuracy (98.88%). Meanwhile, 100 ms is the upper limit of acceptable latency in the touching task [8, 24, 25].

4.5.1 Can our model work with fewer sensors? The pressure signal on the touchscreen is helpful for unintentional touch identification. However, most touchscreen devices have no pressure sensor yet, while a few devices have four pressure sensors in the corners (e.g., the force touch trackpad on MacBook). To explored the feasibility of preventing unintentional touch on existing devices, we evaluated the TypeBoard in three sensor settings.

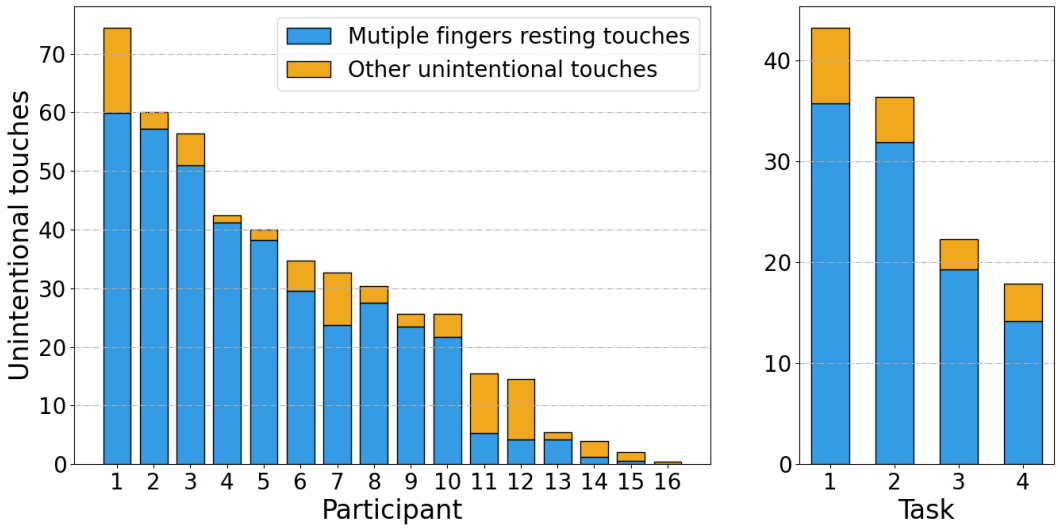


Fig. 8. The unintentional touches per 100 keystrokes over participants and tasks. The blue bars represent those touches that caused by multiple fingers resting, while the orange bars represent other unintentional touches.

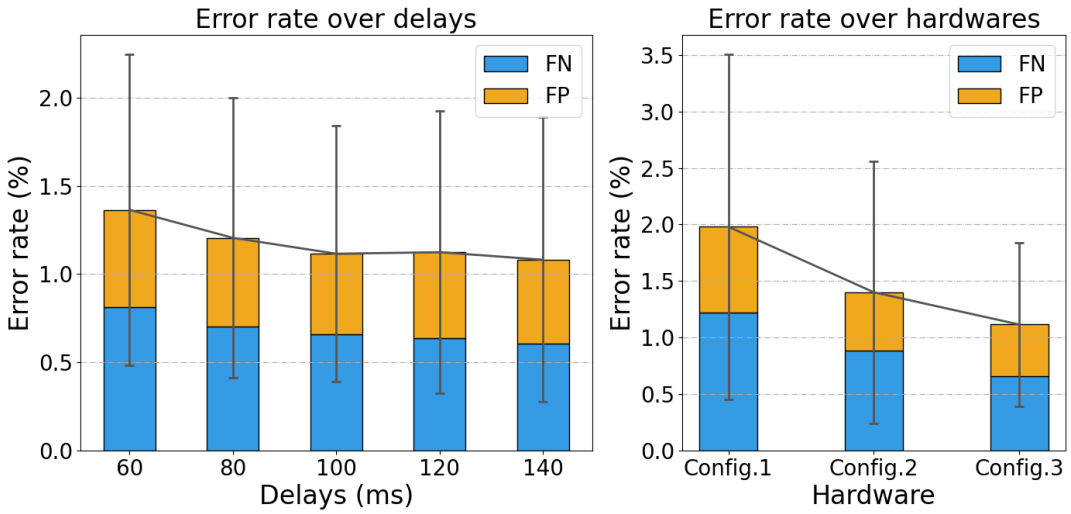


Fig. 9. The recognition error rates over delays and sensor abilities. Error bars indicate standard deviation.

(1) **Capactive touchscreen:** The commonly used touchscreen devices have capacitive signals but not pressure signals. To evaluate our method on these devices, we removed all the features referred to pressure signals and retrained the model.
(2) **Force Touch trackpad:** The MacBook's Force Touch trackpad has four pressure sensors in the corners. The trackpad provides the total pressure on the whole touchpad. We evaluated our model in this setting by a simulation, where we estimated the pressure of each touch as the product of the total touchscreen pressure and the contact area proportion of the touch.
(3) **Pressure-sensitive touchscreen:** Future touchscreen devices may provide high-resolution pressure signals, which is the experimental setting in our paper.

Figure 9(right) shows our method's performances in the three sensor settings. An RM ANOVA shows a significant effect of setting on the recognition accuracy ($F_{2,28} = 10.52, p < 0.001$). Bonferroni-corrected post hoc tests showed significant differences between the following setting pairs: 1-2 (p<.005), 1-3 (p<.005). The difference between setting two and three was a tendency (p=.062). Results show that the touchscreen devices with the total pressure signal strike a balance between recognition rate and hardware cost.

5 STUDY 3: EVALUATION ON TYPEBOARD

The motivations of study three were two-fold. First, we compared the performance and user experience of the TypeBoard and the ordinary touchscreen keyboard. Second, as we introduced in related work, tactile landmarks on keyboards improve users' typing speed by enabling touch typing, so we investigated the feasibility of TypeBoard plus tactile landmarks in this study. In summary, we evaluated users' typing performance on three settings: (1) ordinary software keyboard, (2) TypeBoard, and (3) TypeBoard plus, which was the TypeBoard plus tactile landmarks.

5.1 Participants

We recruited 15 participants from the campus (aged from 19 to 26, M = 20.87, SD = 2.42, seven females). All the participants were right-handed and did not take part in the previous studies. They have used software keyboards on smartphones for not less than two years (M = 6.67, SD = 2.06). Eleven participants have ever used software keyboards on tablets.

5.2 Design and Procedure

The study followed a within-subject design to compare users' typing speed in three keyboard configurations. The participant sat on an office chair. He could adjust the chair to a comfortable position. The participant typed on the pressure-sensitive touchpad to input words and received visual feedback from the tablet. As figure 10 shows, there were three keyboard settings in the experiment as follows:

(1) **Config. 1):** *Ordinary Keyboard.* On the ordinary software keyboard, all contacts on the touchscreen are recognized 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 touch rejection. The system only recognized intentional touches as keystrokes. Users can rest their hands on the keyboard.
(3) **Config. 3):** *TypeBoard plus.* TypeBoard plus refers to the TypeBoard plus tactile landmarks. To provide tactile landmarks on TypeBoard, we attached 0.05 mm thick stickers on the touchpad to simulate physical keys. There were small bumps on the F and J keys, which is the same as the physical keyboard. Users could align their fingers without visual attention.

There were five sessions for each of the three keyboard configurations. In each session, participants transcribed a Chinese paragraph in a Microsoft Word document. There were roughly xx Chinese characters in a task paragraph. We randomly selected the task paragraphs in a typing speed measurement website [1]. Participants were asked to

Fig. 10. The three keyboard settings in study three.

input as fast and accurately as possible. The transcription task is widely used in text entry researches [38, 46, 54] to evaluate the ceiling typing speed. We counterbalanced the order of keyboard configuration using a balanced latin square. Participants had five minutes to warm up before they used each keyboard. They transcribed a paragraph to get familiar with the keyboard. The task phrases in the training step would not appear in the formal experiment. Participants rested for five minutes between sessions to avoid fatigue. On average, participants spent 90 minutes completing the experiment.

5.3 Reslut

A Repeated Measures (RM) ANOVA was conducted for text entry speed, Uncorrected Error Rate (UER), and Corrected Error Rate (CER). The within factors were the keyboard and the session. As UER and CER violated the normalcy, we used the Aligned Rank Transform [51] for correction. If any independent variable had significant effects ($p < 0.05$), we used Bonferroni-corrected post hoc tests for pairwise comparisons.

5.3.1 Speed. We measured text entry speed in Chinese characters per minute (CPM). Participants used Pinyin [3], a phonetic spelling system in Roman characters to input Chinese characters. To input a Chinese character, users type the Pinyin of the desired character (2 - 6 letters) and then select the target from a candidate list. Participants can also type the Pinyin of a Chinese word, consisting of two to four Chinese characters, and then select the whole word. In short, the process of inputting a Chinese character/word is similar to inputting an English word with word prediction/correction. We measured typing speed in CPM with this formula:

$$CPM = \frac{|S|}{T} \times 60 \tag{1}$$

where |S| is the length of the transcribed paragraph in characters (including punctuation), and T is the completion time, i.e., the elapsed time in seconds from the first intentional touch to the last one. All the time consumption, including the time of candidate selection, was taken into account.

Figure 11 shows the users' typing speeds over sessions. There is no significant effect of session on speed ($F_{4,56} = 1.76, p = .15, \eta_p^2 = 0.11$). The result indicates that the learning costs of the three keyboards were low. Participants reached the ceiling performance after a five-minute training. Keyboard has a significant effect on speed ($F_{2,28} = 26.76, p < .001, \eta_p^2 = 0.66$). Pair-wise comparisons show significant differences between all the keyboard pairs: ordinary keyboard vs. TypeBoard ($p < .005$), ordinary keyboard vs. TypeBoard plus ($p < .001$), and TypeBoard vs. TypeBoard plus ($p < .005$). The participants' average typing speed on the ordinary keyboard was 43.71 CPM (SD = 6.52). The typing speed on the TypeBoard was 48.87 CPM (SD = 10.14), outperforming the ordinary keyboard by 11.78%. The typing speed on the TypeBoard plus was 52.97 CPM (SD = 9.85), outperforming

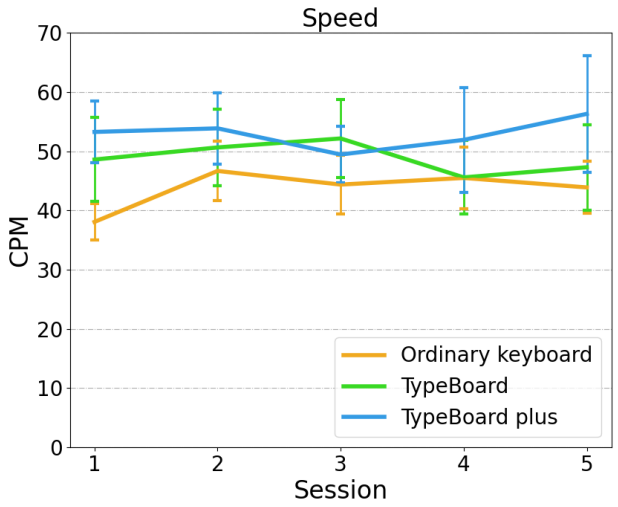


Fig. 11. Text entry speed of the three keyboards over sessions. Error bars indicate 95% confidence interval.

the ordinary keyboard by 21.19%. Results show that the TypeBoard improves the efficiency of the touchscreen keyboard.

5.3.2 Error rate. We used two metrics to measure text entry accuracy: (1) Uncorrected Error Rate (UER) - text entry errors that remain in the transcribed string. UER is the number of uncorrected erroneous Chinese characters divided by the number of correct and erroneous characters. (2) Corrected Error Rate (CER) - text entry errors that are fixed (e.g., backspaced) during entry. CER is the number of corrected erroneous Chinese characters divided by the number of correct and erroneous characters. The corrections of Pinyin while inputting a word were not taken into account of CER. As UER and UER violated the normalcy, we used the Aligned Rank Transform for nonparametric factorial analysis [51].

Figure 12 shows the CER and the UER over sessions. There is no significant effect of session on CER ($F_{4,56} = 1.01, p = .39$). Keyboard has a significant effect on CER ($F_{2,28} = 9.49, p < .005$). Pair-wise comparisons showed significant differences between the following keyboard pairs: ordinary keyboard vs. TypeBoard ($p < .01$), and ordinary keyboard vs. TypeBoard plus ($p < .005$). The average CERs of the ordinary keyboard, TypeBoard, and TypeBoard plus were 6.66% (SD = 4.42%), 4.58% (SD = 3.58%), and 4.21% (SD = 2.58%). There is no significant effect of session ($F_{4,56} = 0.41, p = .71$) or keyboard ($F_{2,28} = 0.001, p = .998$) on UER. The average UERs of the ordinary keyboard, TypeBoard, and TypeBoard plus were 1.29% (SD = 1.67%), 1.28% (SD = 1.38%), and 1.28% (SD = 1.16%). Results show that the TypeBoard reduces the probability of making typos compared with the ordinary keyboard. This is the main reason the TypeBoard improves the typing speed of the touchscreen keyboard. The TypeBoard plus does not reduce the probability of making a typo compared with the TypeBoard. Thus, there are other reasons for faster typing on the TypeBoard plus.

5.3.3 Time components. To gain deeper insight into the performance comparison among keyboards, we broke down the text entry time into four components: typing time, selecting time, deleting time, and pause time. Typing time was the time spent on typing Pinyin. Selecting time was the time spent on selecting a candidate Chinese character/word. Deleting time was the time spent on deleting characters. Pause time span from inputting the last

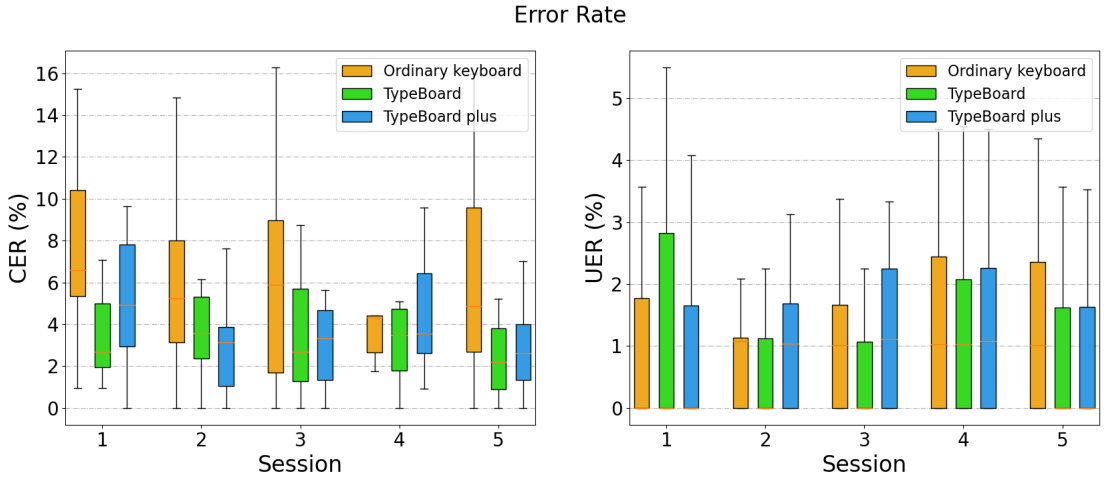


Fig. 12. Uncorrected error rates and Corrected error rates of the three keyboards over sessions.

character/word to starting the next character/word. Figure 13 shows the typing time, selecting time, deleting time, and pausing time per Chinese character over the three keyboards. RM ANOVAs showed that keyboard has significant effects on selecting time ($F_{2,28} = 7.85, p < .005$), deleting time ($F_{2,28} = 20.89, p < .001$), and pause time ($F_{2,28} = 12.76, p < .001$). Pair-wise comparisons showed that both the TypeBoard (p<.001) and the TypeBoard plus (p<.001) reduce the ordinary keyboard's deleting time. The results showed that unintentional touch prevention reduced typos. The TypeBoard plus significantly reduced the pause time compared to the ordinary keyboard (p<.001) and the TypeBoard (p<.005). The results indicated that participants performed touch typing more frequently on the TypeBoard plus, saving the time of finding the next transcribing word.

5.3.4 Touch position. Figure 14 shows the distribution of intentional touches on the TypeBoard and the TypeBoard plus . For convenience, we assumed that the point clouds obeyed 2D Gaussian distributions. While the key width is xx mm and key height is xx mm, the average X/Y offsets were xx/xx mm on the TypeBoard and xx/xx mm on the TypeBoard plus. This result shows that users tend to touch shift towards the xx of the keyboard. The average X/Y standard deviations of the distributions were xx/xx mm and xx/xx mm on the two keyboards. An paired sample t-test shows that keyboard has a significant effect on the average product of X and Y standard deviations (F, p), which indicated that users typed more accurately on the TypeBoard plus (p<). Users touched the tactile landmarks on the TypeBoard plus to aligned their fingers, which improved the typing accuracy.

5.3.5 Subjective Rating and Feedback. Participants rated the subjective speed, accuracy, fatigue, and cognitive load on a 7-point Likert scale (1 - the worst; 7 - the best) after using each keyboard. Figure 15 shows the results. Wilcoxon Signed-Rank tests show that the TypeBoard significantly improves the ordinary keyboard's subjective speed ($Z = -2.27, p < .05$), accuracy ($Z = -3.24, p < .005$), fatigue ($Z = -2.84, p < .005$), and cognitive load ($Z = -1.99, p < .05$). The TypeBoard plus improves the ordinary keyboard's subjective speed ($Z = -3.17, p < .005$), accuracy ($Z = -3.52, p < .001$), fatigue ($Z = -3.34, p < .001$), and cognitive load ($Z = -2.28, p < .05$). The TypeBoard plus is better than the TypeBoard on subjective speed ($Z = -2.40, p < .05$) and fatigue ($Z = -2.85, p < .005$). Results show that both the TypeBoard and the TypeBoard plus improve the ordinary keyboard's user experience.

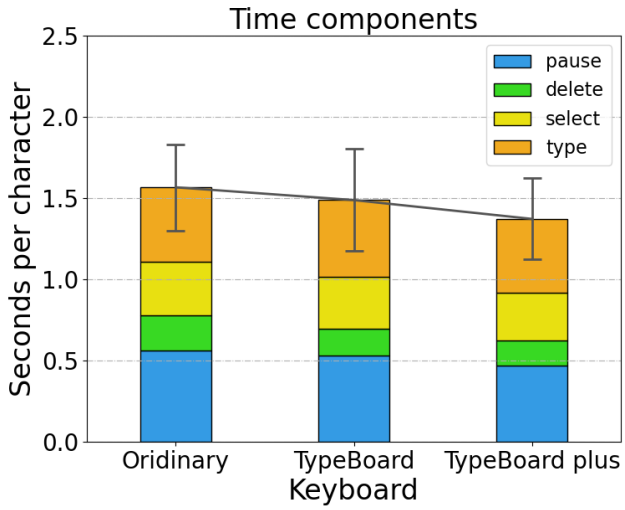


Fig. 13. Time components.

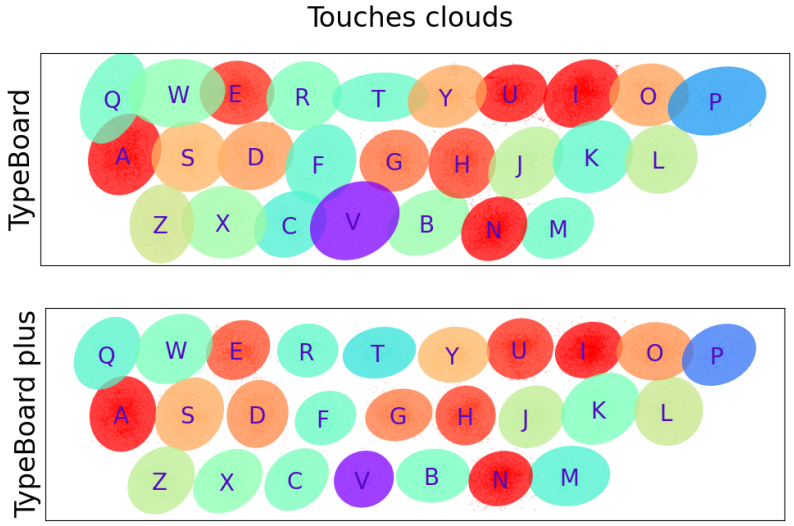


Fig. 14. The distributions of intentional touches on the TypeBoard (above) and the TypeBoard plus (below). The ellipses show 1 standard deviation of the distribution.

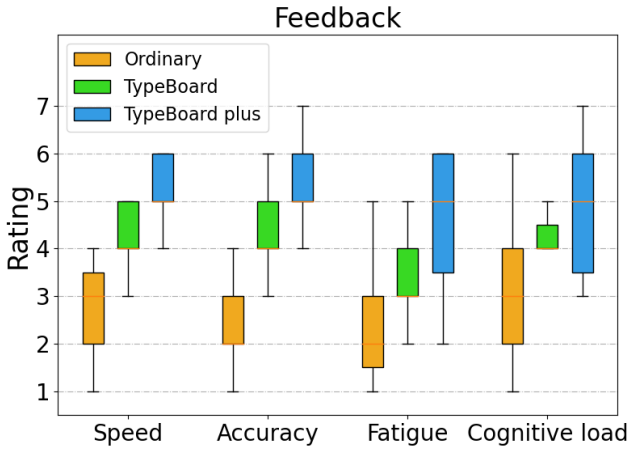


Fig. 15. Subjective ratings (higher is better).

5.4 Summary

5.4.1 Ordinary Keyboard vs. TypeBoard. The TypeBoard improves the ordinary tablet keyboard's typing speed by 11.78%. The TypeBoard has the advantages of avoiding fatigue, relieving cognitive load, and reducing typos.

5.4.2 TypeBoard vs. TypeBoard plus. The TypeBoard plus further improve the TypeBoard's typing speed by 8.51%, outperforming the ordinary tablet keyboard by 21.19%. Compared to the TypeBoard, the TypeBoard plus has the advantages of improving typing accuracy and reducing pause time.

6 DISCUSSION

6.1 Why not deep learning?

In this paper, we used classical machine learning methods (SVM) to solve the problem. We did not use deep learning for two reasons. First, the accuracy of our model was high, nearing the ability of humans. A more sophisticated method can hardly surpass our proposal. Second, the prevention of unintentional touch is a basic and underlying function on touchscreen devices, requiring fast and low-power solutions. Deep learning, as a computationally intensive tool, does not meet the requirement. For these reasons, we argue that classical machine learning methods are more practical for the problem.

6.1.1 The iterative method. A lightspot of this paper is the iterative process to solve the problem, i.e., we developed a semi-finished TypeBoard, then conducted user experiment on it, and finally improved the technique by using the latest dataset. Because the relationship between a technique and the user behavior on it is a "chicken and egg" problem, most previous studies explored user behaviors through experiments on devices with no feedback (like our study one) [6, 31, 40] or substituted feedbacks [29, 52]. We argue that the iterative method deserves more attention. In our work, the user behaviors we observed in study two (with feedback) are different from those in study one (without feedback). The model trained by the latest dataset also performed better. That is, the iterative process improved our technique and helped to gain a more practical model of user behavior.

6.2 Other ways to improve the detection?

First, we can leverage the keyboard layout as a basis for unintentional touch detection, e.g., when a touch does not fall on any button, it has a greater probability of being an unintentional touch. Second, we can use the language model as a priori knowledge. The Bayesian decoder is widely used to predict users' desired words from the vocabulary [18, 20, 23, 39]. The decoder can also calculate the probability distribution of the subsequent touch. When a touch falls on the low probability button, it is more likely to be an unintentional touch. Our proposal leveraged neither the keyboard layout nor the language model because we explored the general method to solve the unintentional touch problem.

6.3 Language dependence of the TypeBoard.

Our studies were conducted in Chinese. To achieve the best performance in other languages, we suggest a reproduction of our study two on the target language, and using (1) the existing feature vector and (2) the new dataset to retrain the model.

(1) *Why the existing feature vector is adequate?* The Chinese input method was comprehensive. We observed various unintentional touch cases in the study, which helps us design the feature vector thoughtfully. There are many keyboard layouts (e.g., English, German, France, and Russian) on which users type directly to enter characters. The orthography used for Chinese or other East Asian languages (e.g., Japanese and Korean) requires special input methods. Users narrowed down the range of possibilities by entering the desired character's pronunciation and then selected the desired ideogram. Generally speaking, user behaviors in Chinese text input tasks are more diverse.
(2) *Why we suggest a reproduction of study on the target language?* In study two, we found that the typing behavior in details (e.g., the frequency of each kind of unintentional touch) significantly impacted the model training results. We believe there is typing behavior difference in different languages, so adapting to the target language should improve the TypeBoard performance.

6.4 The TypeBoard plus vs. touchscreen overlays.

The TypeBoard plus enables touch typing on tablets by allowing users to rest their fingers on touchscreens. There are other solutions to support the finger resting on the touchscreen. TouchFire [48], SLAP Widget [50], and the Sensel Morph [4] offer an overlay on the touchscreen. Only keystrokes are transferred to the touch screen by a mechanical structure, and other touches are blocked. The TypeBoard plus has two advantages compared to the touchscreen overlay solution. First, we can add tactile landmarks on the TypeBoard through built-in devices such as deformable screens [2] and changeable surface texture [5, 9, 33], while the touchscreen overlay is an external object. Second, the tactile landmarks on the TypeBoard were only 0.05 mm thick. Users can perform text input, and cursor control in the same space [28], which reduces task-switching costs and improves efficiency. In contrast, the touchscreen overlay bumps are much thicker, preventing users from using the touchscreen as a trackpad.

6.5 Other ways to improve the user experience?

First, in the experiment, we provided audio feedback for each keystroke. Previous work showed that the tactile feedback of a click could also improve the accuracy and speed of typing [37]. In the future, we can use vibration to simulate physical buttons' tactile feedback, similar to the MacBook Trackpad. Second, the TypeBoard users cannot directly press keys with their fingers resting. We should enable this feature in the future.

7 CONCLUSION

一些结论。

REFERENCES

[1] 2011. Typing speed measurement. http://www.51dzt.com/test/
[2] 2020. Tactus technology. http://www.tactustechnology.com/
[3] 2021. Pinyin. https://en.wikipedia.org/wiki/Pinyin
[4] 2021. The Sensel Morph. https://morph.sensel.com/
[5] 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.
[6] 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.
[7] 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.
[8] Christiane Attig, Nadine Rauh, Thomas Franke, and Josef F Krems. 2017. System latency guidelines then and now–is zero latency really considered necessary?. In *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 3–14.
[9] 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.
[10] Matthew N Bonner, Jeremy T Brudvik, Gregory D Abowd, and W Keith Edwards. 2010. No-look notes: Accessible eyes-free multi-touch text entry. In *International Conference on Pervasive Computing*. Springer, 409–426.
[11] Jörg Brakensiek and Raja Bose. 2013. Method and apparatus for precluding operations associated with accidental touch inputs. US Patent App. 13/221,344.
[12] B Chaparro, B Nguyen, M Phan, A Smith, and J Teves. 2010. Keyboard performance: iPad versus Netbook. *Usability News* 12, 2 (2010), 1–9.
[13] Matthew JC Crump and Gordon D Logan. 2010. Warning: This keyboard will deconstruct—The role of the keyboard in skilled typewriting. *Psychonomic bulletin & review* 17, 3 (2010), 394–399.
[14] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.
[15] Olumuyiwa M Durojaiye and David Abzarian. 2016. Classification of touch input as being unintended or intended. US Patent 9,430,085.
[16] 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.
[17] Leah Findlater, Jacob O Wobbrock, and Daniel Wigdor. 2011. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2453–2462.
[18] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.
[19] Andrew M Gordon and John F Soechting. 1995. Use of tactile afferent information in sequential finger movements. *Experimental brain research* 107, 2 (1995), 281–292.
[20] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3817–3821.
[21] Gregg S Goyins and Mark F Resman. 2001. Palm pressure rejection method and apparatus for touchscreens. US Patent 6,246,395.
[22] Jason Tyler Griffin. 2013. Touch screen palm input rejection. US Patent App. 13/469,354.
[23] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[24] Topi Kaaresoja, Stephen Brewster, and Vuokko Lantz. 2014. Towards the temporally perfect virtual button: touch-feedback simultaneity and perceived quality in mobile touchscreen press interactions. *ACM Transactions on Applied Perception (TAP)* 11, 2 (2014), 1–25.
[25] Topi Johannes Kaaresoja. 2016. *Latency guidelines for touchscreen virtual button feedback*. Ph.D. Dissertation. University of Glasgow.
[26] Terho Kaikuranta. 2006. Method for preventing unintended touch pad input due to accidental touching. US Patent 6,985,137.
[27] Jeong Ho Kim, Lovenoor Aulck, Michael C Bartha, Christy A Harper, and Peter W Johnson. 2014. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied ergonomics* 45, 6 (2014), 1406–1413.
[28] 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.
[29] 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.
[30] Seiya Koura, Shunsuke Suo, Asako Kimura, Fumihisa Shibata, and Hideyuki Tamura. 2012. Amazing forearm as an innovative interaction device and data storage on tabletop display. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*. 383–386.

[31] 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.

[32] 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.

[33] 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.

[34] Gordon D Logan and Matthew JC Crump. 2011. Hierarchical control of cognitive processes: The case for skilled typewriting. In *Psychology of learning and motivation*. Vol. 54. Elsevier, 1–27.

[35] 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.

[36] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–24.

[37] Zhaoyuan Ma, Darren Edge, Leah Findlater, and Hong Z Tan. 2015. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. In *2015 IEEE World Haptics Conference (WHC)*. IEEE, 220–227.

[38] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. 754–755.

[39] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1073–1082.

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

[41] 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.

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

[43] 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.

[44] 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.

[45] Andrew Sears. 1991. Improving touchscreen keyboards: design issues and a comparison with other devices. *Interacting with computers* 3, 3 (1991), 253–269.

[46] R William Soukoreff and I Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 113–120.

[47] Hussain Tinwala and I Scott MacKenzie. 2010. Eyes-free text entry with error correction on touchscreen mobile devices. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. 511–520.

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

[49] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. 2014. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 14–27.

[50] 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).

[51] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 143–146.

[52] 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.

[53] Teppei Yajima and Hiroshi Hosobe. 2018. A Japanese Software Keyboard for Tablets that Reduces User Fatigue. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 339–346.

[54] Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi, and Yuanchun Shi. 2017. Word clarity as a metric in sampling keyboard test sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4216–4228.

[55] 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.