The Tap Strap 2: Evaluating Performance of One-Handed Wearable Keyboard and Mouse

Kristian Mrazek
Dept. of Math and Computer Science
Augustana College
Rock Island, United States
kristianmrazek17@augustana.edu

Brian Holton
Dept. of Math and Computer Science
Augustana College
Rock Island, United States
brianholton19@augustana.edu

Tanner Klein
Dept. of Math and Computer Science
Augustana College
Rock Island, United States
tannerklein20@augustana.edu

Izan Khan
Dept. of Math and Computer Science
Augustana College
Rock Island, United States
izankhan18@augustana.edu

Thomas Ayele
Dept. of Math and Computer Science
Augustana College
Rock Island, United States
thomasayele17@augustana.edu

Tauheed Khan Mohd
Dept. of Math and Computer Science
Augustana College
Rock Island, Illinois
tauheedkhanmohd@augustana.edu

Abstract—This paper presents an evaluation of Tap Strap 2, a one-handed wearable keyboard that uses tap combinations to type characters. The purpose of the Tap Strap 2 is to give the user a portable keyboard that is easier to use for mobile devices than the on-screen QWERTY keyboards included on those devices. The user taps different combinations of fingers on a surface to enter different characters and perform a variety of commands. By using the embedded sensors in each ring, the Tap Strap 2 can recognize which fingers are tapping at any given time. Tap Strap 2 also has a mouse mode that can be switched to from keyboard mode, allowing the user to browse websites on a mobile device as they would on a computer. The third mode that Tap Strap has is a gesture mode that can be used for mobile devices as well as in virtual reality settings. This article details what Tap Strap 2 is capable of and describes the hardware. It also provides an evaluation of the device's usability and tests an auto-correct algorithm meant to reduce the probability of spelling errors when using the Tap Strap 2.

Index Terms—keyboard, wearable, tap, mobile, touch

I. INTRODUCTION

As computer systems evolve, so do methods of human-computer interaction (HCI), particularly interfaces and the means through which input, such as text that the user wishes to type, is provided to the computer. The traditional QWERTY keyboard, in both its physical and onscreen forms, remains the dominant means of inputting text at the time of this paper. Unfortunately, the on-screen QWERTY keyboards on mobile devices can prove problematic to use due to the compact layout of the keys, which encourages spelling errors. These keyboards are also very difficult to use for the visually impaired, since the keys are small and flat. Currently, the solution to these issues is the implementation of a text-to-speech function on phones and smartwatches. However, even this has significant issues, as if the user does not speak clearly or there is too much noise in the surrounding environment, the device will incorrectly interpret the words and fail to produce the desired output.

As a potential solution to these problems, Tap Systems, Inc. introduced a wearable keyboard called the Tap Strap 2. Tap Strap 2 is a five-ringed device that can slip onto a hand and connect to a device using Bluetooth. The user slides each ring onto each finger, with the largest ring going on the thumb. Attached to the thumb ring are the power button and the mouse portion of the device, as shown in Figure 1.

The Tap Strap 2 has three modes: the default keyboard mode, mouse mode, and air gesture mode. While in keyboard mode, the user can type by tapping different combinations of fingers at the same time on a solid surface. Taps are detected through the use of built-in accelerometers in each ring. This information is then sent to and processed by a microcontroller unit (MCU) in the thumb ring, determining the input. As is shown in Figure 2, each combination corresponds to either a command key, such as tab or backspace, or a character key. For example, each vowel is created by tapping a single finger, beginning with "a" at the thumb and ending with "u" at the small fifth finger. Consonants are typed using two or more different fingers, and special characters, such as punctuation marks, are typed by tapping the same combination two or three times in rapid succession. Certain letters, such as "j" and "q", have difficult combinations that require the user to hold up their ring finger while tapping the two adjacent digits. To remedy this, shortcuts are implemented for these letters that require the player to simply tap one or two fingers twice in quick succession, similarly to special characters. The learning curve of the device is quite steep, as the user needs to memorize a brand new keyboard layout with many seemingly random combinations, though some of them, such as the vowels, are fairly straightforward to memorize. Some of the tapping combinations themselves can also be physically demanding to execute for individuals who suffer from low hand dexterity.

The Tap Strap 2's mouse mode works through the use of an optical chip embedded in the thumb ring. The user can activate
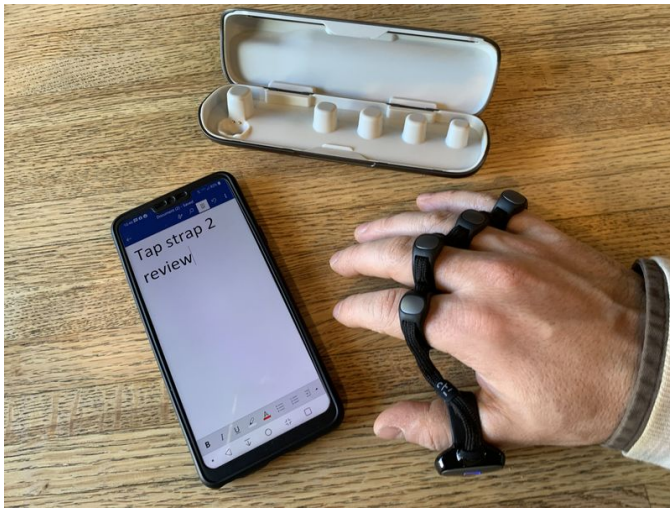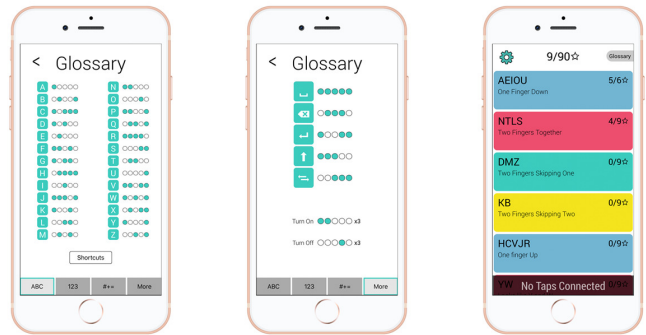
Fig. 1: The Tap Strap 2 and its case.

Fig. 2: The Tap Strap 2's glossary, accessed through the TapGenius app. Green circles indicate the fingers that are tapped to type a given character.

mouse mode by resting their thumb on a surface. The cursor can then by guided by dragging the thumb across the surface in the direction the user wishes to move it. There are also several mouse commands, such as clicking and scrolling, that can be done by tapping fingers when in mouse mode. Mouse mode is disabled by simply lifting the thumb off the surface.

The third and final mode that this technology has is the air gesture mode. This is done through the use of a 6-axis inertial measurement unit, or IMU, which is built into the thumb ring. This unit is used in tandem with the accelerometers to determine the position of the hand in the air. Air gesture mode is triggered when the user holds their hand out in what the Tap Strap 2's official site refers to as a "handshake" position. By default, the air gesture mode turns the Tap Strap 2 into a mouse, where the user can direct the cursor with their hand and perform mouse commands (clicks, scrolling, etc.) by extending fingers and flicking them quickly in different directions. In addition to computers and mobile devices, the air gesture mode can also be used with Bluetooth television sets and media players.

In addition to the included manual and glossary, there is also an app called TapGenius, which is created to help the user learn the Tap Strap 2's inputs. TapGenius was designed with the help of Learning Neuroscience specialists at Stanford University. TapGenius teaches the user how to learn the Tap Strap 2's keyboard "layout," beginning with simple inputs such as vowels and later progressing to the more complicated consonants and punctuation marks. The training app breaks the alphabet into eight groups, ranked in order of difficulty. The app also teaches the user how to type words that employ characters that the user has already learned. The TapGenius app also gives the user a vibration in the thumb ring whenever an incorrect character is entered.

II. RELATED WORK

The past decade has seen the rise of wearable devices as a major force in technology and HCI. These "wearables"

are a form of ubiquitous computing in which computing is embedded in seemingly mundane objects, with the most prominent of these being wristbands and smartwatches [1]. Many consumers across the globe have shown a strong interest in wearables. As a result, researchers and industry leaders seek new and innovative ways of interacting with daily devices, including smartphones, smart TVs and personal computers, besides the standard QWERTY keyboard. Moreover, the rise of augmented reality (AR) and virtual reality (VR) technologies in the computing and gaming world demands the existence of an input system that is based entirely on touch and gestures rather than eyesight [2].

A. An Overview of Wearables

Almost a decade ago, the North American PC gaming market saw a new addition in the form of the Peregrine Glove [3]. The glove comes with 18 touch zones that enable the user to communicate with any application where a keyboard is needed. With just a touch of the user's fingers, the user could perform any command they wanted to.

In 2013, Jing et al. [4] proposed the Magic Ring, a hands-free input device designed to recognize finger gestures. This device uses inertial sensors and can be used as a remote controller for electrical appliances such as televisions and radios. It does this by using an accelerometer to recognize gestures, much like the Tap Strap 2, then outputting an infrared signal that interacts with the object of interest. The Magic Ring also has the ability to accurately recognize when the user is performing a variety of day-to-day activities such as running, cooking, and writing. In tests performed with the Magic Ring, the rate of successful recognition was 97.2% [4].

In 2014, a system named Airwriting was developed at Germany's Karlsruhe Institute of Technology [5]. This used data from an IMU, accelerometer, and gyroscope, all of which were attached to the back of the user's hand, to detect gestures that correspond to the characteristic signal pattern for every alphabet letter. The gestures are then translated into editable text for phones, tablets, and other mobile devices. Tests run

using this system yielded an average spelling error rate of 11%.

In 2012, Southern et al. [6] introduced a system called BrailleTouch. This is used to help blind people type on mobile devices, which is a challenge since most of these devices do not have physical keyboards attached to them. The device is set up in a way so that the blind user holds a phone facing away from them, with three fingers on each hand resting above dots on either side of the touchscreen. It is remarkably similar to the Tap Strap 2 in a sense, since the user types characters by tapping their fingers on the touchscreen using the same patterns as the Braille system. BrailleTouch was a much more affordable alternative to the comparable Braille Sense Plus system, which costs $6000.

TypeInBraille, another system that had emerged the previous year with the efforts of Mascetti et. al. [7], similarly employed tap combinations. However, it generates all three rows of each Braille character separately. There are commands to raise both the left and right dots in each row as well as to move on to the next character. This proved to be both more accurate and efficient than the onscreen QWERTY keyboard, with 55% higher accuracy than the QWERTY in a noisy "tramcar" setting. Both BrailleTouch and TypeInBraille prove to be significant to this research, since while the Tap Strap 2 can prove useful for the blind due to relying on touch rather than sight, it suffers from a notable hindrance in that it requires them to learn a new alphabet rather than simply using the existing Braille system.

In 2015, Nirjon et al. [8] introduced the TypingRing, a wearable system in the form of a "ring" that can be slid onto the user's middle finger, enabling them to type as if an invisible standard QWERTY keyboard is lying on a surface underneath their hand. A user study states that Typing Ring yields a reasonable typing speed (e.g., 33–50 characters per minute), which improves over time as the users become more acquainted with the device.

In 2016, Gordon et. al. [9] proposed the WatchWriter, a keyboard specifically designed for smart watches. Users can either use one tap per character or one gesture per word. This device helps since smartwatches are the next device in portable devices, and it can be difficult to accurately type out a message on one because of its size. The makers of WatchWriter decided to try to either optimize or change the current keyboard to help with the small size of the watch, while also making it easier for people to use the WatchWriter.

Wearable computing clearly improves the experience the user gets from technological devices, especially for those with sensory impairments. These devices will keep improving as more companies invest in research and innovation in the field. The wearables of the future will not only continue to make computing more ubiquitous, but they will be more user-friendly as well.

B. Comparing the Tap Strap 2 to Previous Technologies

Most consumers frequently employ information services from cellular towers and terminals to conduct various activities (e.g., checking e-mails, finding a route, texting, dialing a number). However, in order to use these services, the consumer has to pull the smartphone out of their pocket, which can be inconvenient. The advent of smartwatches did little to help. The available options in the market can provide data-driven notifications and timely updates; however, the smartwatch's small screen size means there is not enough space for solid gesture commands [9]. Therefore, the primary input method for a smartwatch is usually voice control. However, there are some technological limitations to this method. Due to the nature of speech recognition, background noises can hinder the device's capability to recognize the command. Moreover, voice control does little to preserve the user's privacy. In the meanwhile, glove-based input systems like the Peregrine Glove and Airwriting cover up the user's hand, which can impede the user in the execution of certain manual tasks. While the Magic Ring may cause little obstruction, it is poorly suited to typing. Most of these devices can replace a traditional keyboard, but using them involves complex gestures that require time and effort to master. Furthermore, if a user performs gestures without any support for the arm, the postures can be very uncomfortable.

The Tap Strap 2 seems better suited to its role than its counterparts. It has a series of embedded sensors that monitor the fingers and return mechanical information, such as acceleration, rotation, and orientation. The sensors then process data, and the finger-tap combinations are transformed into characters for output. The device is placed at the top of the fingers (close to the knuckles) and engages all five digits at the same time. Despite its user-friendly design and working mechanism, using "the Tap Strap" significantly constrains the user's interaction with the surroundings, though not to the extent of a glove. In comparison to this, the TypingRing can fulfill the user's typing requirements with minimal obstruction, but it requires the user to employ an extensive hard surface to type on (such as a wall or a table).

When compared with the original Tap Strap, the Tap Strap 2 comes with a new thumb ring glider and a more sensitive mouse functionality than its predecessor. When the first iteration of the Tap Strap was released, Tap stated that any surface could be used for typing. It was supposed be able to type virtually anywhere, as some individuals using it are traveling, leaving them with no solid surface to type on. However, following the release of the Tap Strap 2, Tap made it clear that any surface that is too soft or irregular may not work as intended.

III. EXPERIMENTAL SETUP

Before the Tap Strap 2 was tested, information regarding the advantages and disadvantages of the device were collected from customer reviews on the Tap Strap 2 website, Amazon, and other review sites. This was done to see if this is truly groundbreaking HCI technology or merely a work in progress with some potential.

It was possible to locate some common complaints regarding the Tap Strap 2. Most of these were related to the

hypersensitivity of the Strap's sensors. The device did not register characters as accurately as many users expected it to. In order to use the Strap, the user has to tap certain fingers and keep the rest stationary. Many encountered difficulty doing this and frequently produced errors. Users complained of typing the correct letter, but having the Strap produce the wrong output. For example, when one user was trying to type an "o", the Strap typed an "s", since the inputs for "o" and "s" are similar. When the consumer tried to fix the error, the Strap toggled the number pad and began to produce unwanted numbers instead.

A. Accuracy of the Tap Strap

a) Hard Surfaces: To test the overall accuracy of the Tap Strap 2, the device was used on hard, soft, and irregular surfaces. The first test was on a hardwood table. The test consisted of typing each letter one hundred times, only deviating from that when a shortcut is available. When a shortcut of the letter were available, fifty regular taps were tested alongside fifty shortcut taps.

b) Soft Surfaces: The same test was then performed on a bed. Lying in bed while using electronic devices is an activity that many people enjoy doing. The test of soft surfaces was performed to see if typing on a bed with the Strap is a plausible option, as well as to see how much accuracy is lost compared to hard surfaces. The test was conducted in the exact same format as the hard surfaces test.

c) Irregular Surfaces: This experiment is focused on the usability of the Tap while traveling. While away from home, there is not always a table available to type on. Often, the only way to use this device in these situations is on one's own leg. Even though Tap Systems, Inc. said that irregular surfaces may result in less accuracy with typing, a test on irregular surfaces was performed to see how much accuracy is lost. This test was performed in the same manner as the tests for hard and soft surfaces.

B. Recognizable Bluetooth

A major interruption noticed while using the Tap, if the device connected to the Tap would go into sleep mode or would turn off, Tap would not reconnect to the device. The test conducted for Bluetooth was to see if it would only disconnect on specific devices, or if each device would need to be manually reconnected. Before experimenting, the inner workings of Bluetooth and how it functions were researched. From the article by Bisdikian [10], Bluetooth uses asynchronous data that flows between the connected devices. This connection, allows the connection between the devices to be reestablished. Different devices used were cell phones, laptops, and computers.

C. Virtual Reality Capability

Virtual reality (VR) capability was another supposed feature of the Tap Strap 2. This feature was considered to be one of the main selling points of the device, since one can use the air gesture mode to interact with the VR setting. To test this feature, the device was hooked up to an Oculus Quest 2 headset. The goal of this was to test the functionality of the Tap Strap 2 in a VR setting.

D. Mouse Functions

Another important feature of the Tap Strap 2 was the built-in mouse. The usability of the mouse on the Tap Strap 2 was a major concern, so a series of tests were set up to test it. The specifications were researched for the mouse function in the Tap Strap 2 and they were compared to those of a regular computer mouse using the patent by Bieber [11]. The mouse in the patent is an optical mouse with laser illumination. The Tap Strap 2 also has an optical mouse with a laser sensor to detect movements. Both the efficiency and accuracy of the mouse will be quantified to determine its overall usability.

a) Efficiency: To test the efficiency of the mouse, it was determined how fast the mouse could scroll down a fifteen-page research paper. The test conducted used three different methods to get to the bottom of the page: the scroll bar on the right-hand side, the built-in scroll down function (mouse wheel), and the scroll lock. Each mouse was tested ten times with each scroll method.

The efficiency of the mouse's 'drag and drop' function was also compared to that of a normal mouse. For this test, a particular word or sentence was highlighted. Since the mouse could not tell when to stop highlighting, two different methods were tried. Method one let the mouse 'drag and drop' time out, the second method was clicking to get rid of the drag and drop.

b) Accuracy: At first use, it was apparent that the Tap mouse was quite sensitive, so it was also important to test the accuracy. To test the accuracy of the mouse on the Tap Strap 2, a test was performed where the user had to click on as many targets as possible in a minute. This was done first using a normal mouse, then with the Tap Strap 2's built-in mouse mode. The test was performed ten times for each mouse.

E. Accuracy of the Tap Strap with an Auto-Correct Feature

A common critique of the Tap Strap 2 is that there is no built-in spell checker or auto-correct feature. This makes sense, as characters are typed by tapping different finger combinations, so the user will find themselves experiencing more input errors with the Tap Strap 2 than a normal QWERTY keyboard. While the auto-correct features on mobile devices will generally catch these errors, this is not the case with normal computers. It is especially problematic when one is using a text editor like Notepad, which has no built-in spell checker, since errors can easily go unnoticed. Spelling errors tend to also be a nuisance to fix, since the backspace command can add more unwanted characters if typed incorrectly.

In order to determine how much a built-in auto-correct feature improves the overall experience of the Tap Strap 2, a simple Notepad-like program was designed in Python, which incorporated three different spell checker modules. This was done in order to ensure that the auto-correct feature had the largest dictionary possible, preventing real words from

getting corrected into other words. For the test, a set of 50 sentences was entered using the Tap Strap 2 into the normal Notepad (without any use of an auto-correct feature). These sentences, borrowed from the list of Harvard sentences, are useful because they provide a reasonable approximation of the frequency of every letter in the English language. Gordon et. al. [9] employed a similar set to test the auto-correct feature for the watchWriter. The sentences were then entered into the Python text editor, which makes it easier to determine how effective the auto-correct feature was.

F. Obtaining Raw Data from the Device

There is a software development kit (SDK) available for the Tap Strap 2 on the TapWithUs GitHub. This contains code that allows the user to collect various forms of information from the device, such as inputs and raw sensor data, provided the Tap Strap 2 is connected to the computer and in developer mode. The raw accelerometer and IMU data is of particular interest here because if trends can be found in the data with respect to the character inputs themselves, it may be possible to propose a machine learning algorithm that will improve the accuracy of the device's readings. Therefore, for this experiment, the program was allowed to collect data in "raw sensors" mode while the alphabet was typed in order four times.

IV. EXPERIMENTAL RESULTS

Tables I, II and III summarize the experiments done on Bluetooth connectivity, mouse functionality, and accuracy of Tap Strap 2's keyboard, respectively. When looking at the Bluetooth connectivity results, it can be seen that it failed to connect to both a desktop PC and a laptop in sleep mode. However, it successfully connected 100 percent of the time to a phone while the phone was in sleep mode and when the phone was restarted. Taking a look at the comparison between a regular mouse and the Tap Strap 2 mouse, it can be seen that the a regular mouse outperforms the Tap Strap 2 mouse in navigation speed, scroll wheel speed, and scroll lock speed. The Tap Strap 2 mouse was also tested for its drag and drop speed and clicking ability. For that experiment, the Tap Strap 2 was able to perform 2 out of 10 drag drops and 5 out of 10 clicks accurately. Moreover, it can also be seen that a regular mouse is approximately twice as accurate as the Tap Strap 2, hitting an average of 41.3 targets out of 60 while the Tap Strap 2 mouse hit an average of only 19.5 targets out of 60.

Figures 3, 4 and 5 summarize Tap Strap 2's accuracy on three types of surfaces: hard, soft, and irregular. The accuracy of the typing depended on the combination of finger taps required to produce a character. From the experiment performed, it was concluded that simple combinations using one finger or consecutive fingers tend to be more accurate than keys such as 'q' and 'w', which require the user to perform the difficult task of lifting the ring finger. In addition to that, it was also found that the letter 'a' had the highest accuracy, while the the letter 'w' had the lowest accuracy. Lastly, the experiment helped identify that the Tap Strap 2 performs better on harder surfaces, with an average accuracy of 84.7%, while it

has a 76.07% average accuracy on soft surfaces and a 80.23% average accuracy on irregular surfaces.

For the VR test, the Oculus Quest 2 headset connected to the Tap Strap 2 for a few seconds and disconnected. When reconnection was attempted, the Oculus failed to recognize the Strap entirely.

When looking the accuracy of the Tap Strap 2 with an Auto correct feature experiment. It was found that after typing a set of a 100 sentences, the Tap Strap 2 had an error rate of 16.39% without the use of auto-correct. In terms of the types of errors, this experiment yielded 90 character insertions, 71 substitutions, and 3 deletions. This shows that the overwhelming majority of errors happen due to the Tap Strap 2 detecting either the wrong input or a second input immediately after the intended one. Deletions, where the Tap fails to register a character at all, are comparatively rare. When auto-correct was used on this text, the error rate decreased to 4.39%. This shows that even a relatively simple auto-correct feature makes a considerable improvement to the accuracy of the device.

Lastly, the graph in figure 6 represents the raw data that was obtained from the Tap Strap 2 using the Tap SDK. In this case, the letter "a" was typed repeatedly by simply tapping the thumb on a table. The taps appear as spikes on the plot, and it is apparent that the accelerometer attached to the thumb ring shows a much higher acceleration than the other rings, as expected. It would be useful to obtain the actual input readings alongside the raw data; that is, determining whether an "a" was actually typed as opposed to some other character. This would assist in training a model to improve the accuracy of the input readings from accelerometer data, which would likely be done through the use of convolutional neural networks (CNNs) [12]. Unfortunately, however, the Tap SDK unfortunately only allows the user to extract either the raw data or the input information at once.

TABLE I: Bluetooth connectivity

	Number of successful trials	Number of trials
Phone (Sleep)	10	10
Phone (Restart)	10	10
Computer(Sleep)	0	10
Laptop (Sleep)	0	10

TABLE II: Regular mouse vs. Tap Strap 2 mouse efficiency

	Navigation Speed (sec)	Scroll Wheel (sec)	Scroll Lock (sec)
Regular	1.94	5.47	1.97
Tap	7.33	29.17	7.01

TABLE III: Regular mouse vs. Tap Strap 2 mouse accuracy

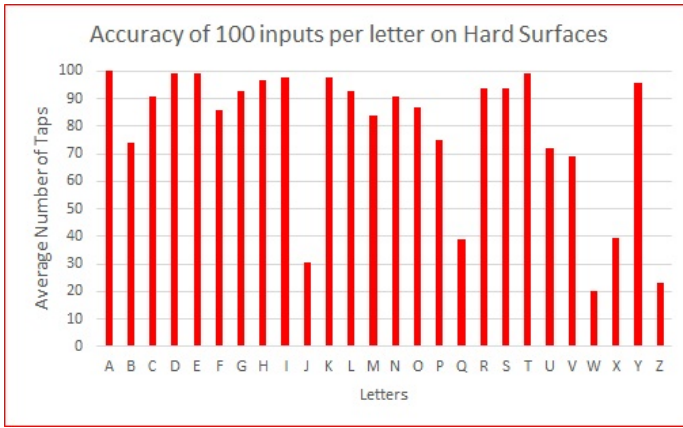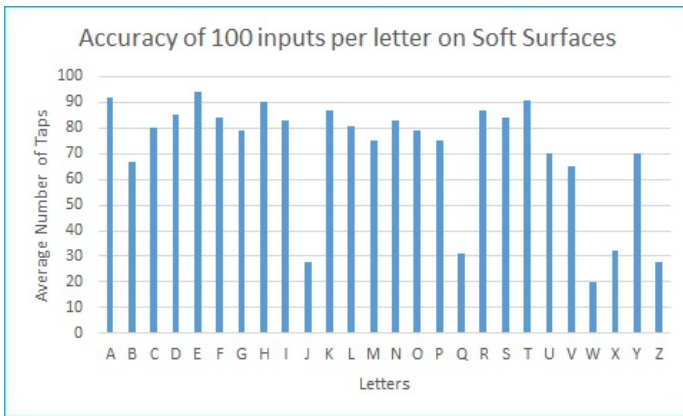|	number of successful trials	of attempts	number of trials
Regular	41.3	60	10
Tap	18.5	60	10

Fig. 3: Accuracy of the Tap Strap 2 keyboard on a hard surface

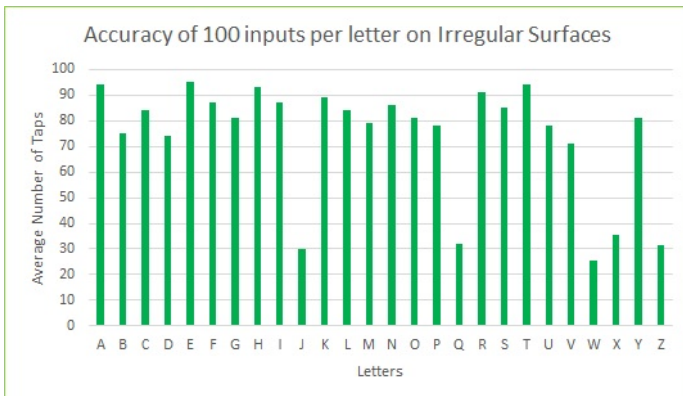Fig. 4: Accuracy of the Tap Strap 2 keyboard on a soft surface

Fig. 5: Accuracy of the Tap Strap 2 keyboard on an irregular surface

Fig. 6: Raw accelerometer data obtained while repeatedly typing the character "a."

It is evident that while using the Tap Strap 2 with an auto-correct mechanic significantly improves the accuracy of words, there are still certain situations in which the auto-correct feature fails. The most obvious example of this is when an input error results in a real English word; for example, typing in "bike" when attempting to enter "bake." Additionally, the program fails to fix even some errors that do not result in
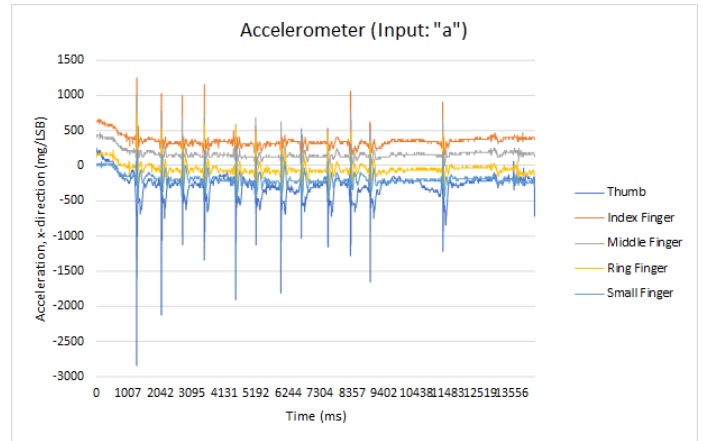
a word, such as accidentally typing "veut" instead of "vest," even though "vest" is in the dictionary and "veut" is not a real word in English. Since the implemented auto-correct feature simply looks for the most common similar words whenever it detects a spelling error, it cannot determine context within a sentence, causing it to miss these kinds of mistakes. The solution to this problem will be to implement a spell checker that uses a recurrent neural network, or RNN.

This RNN will be comparable to a nested RNN system developed by Li et. al. [13]. This system consists of a character-level RNN, which examines the series of characters within a word and finds real words with similar spellings, nested within a word-level RNN, which uses the structure of a sentence to determine which word should be present. The result is a neural network system that focuses on both the spelling of a word and the surrounding context. It is remarkable due to its ability to reliably detect both real and non-real word errors, especially when compared to previous RNN-based and non-RNN-based spell checkers. Furthermore, this research has an interesting way of generating pseudo training data, using a confusion matrix that maps characters (or groups of characters) to other characters that represent phonetic sounds. High confusion coefficients correspond to characters or groups of characters that are often confused in spelling. The pseudo training data is then generated through random perturbations of real dictionary words.

It is worth noting that instead of "normal" RNNs, the work of Li et. al. employed gated recurrent units (GRUs) [13]. The GRUs were used to prevent the "vanishing" and "exploding" gradient problems [14]. These are serious problems that result from a rapid spike or decline in error gradient values, and they cause the RNN to enter a highly unstable state and possibly cease to be able to learn entirely [15].

The key difference with the model that will be used here has to do with the nature of the perturbations themselves. The issues with the Tap Strap 2 occur not from the user confusing sounds, but from inputs being misinterpreted. The character-level matrix will be created to train the RNN to find

spelling errors that are likely to occur with the Tap Strap 2. For example, since "a" (thumb only) and "n" (thumb and index finger) have similar inputs, they will share a high confusion coefficient; however, "a" will share a low confusion coefficient with "e" (small finger only).

V. Conclusions and Future Work

In this research, the overall usability of the Tap Strap 2 was investigated and evaluated with a particular focus on accuracy and efficiency. The device was able to consistently connect to a phone in sleep and restart mode, but failed to do so with laptop and desktop PCs. The device's mouse functionality was surpassed by a normal mouse in its ability to scroll quickly and accurately click targets. The Tap Strap 2 keyboard's accuracy was first found to be 84.7% when used on a desk, 76.07% when used on a bed, and 80.23% when used on a leg. Therefore, it can be stated with a reasonable level of confidence that the Tap Strap 2 functions nearly as well on soft or irregular surfaces as on flat, hard surfaces. However, the device was found to be unable to stay connected to a VR headset for more than a handful of seconds, which is a critical error since the device was marketed as being useful in VR settings. A simple Python-based spell checker was found to drop the word error rate of the Tap Strap 2 from 16.39% to 4.39%. Finally, an RNN-based machine learning model was presented with the goal of improving the spell checker's accuracy.

Hence, the Tap Strap is still surpassed by the traditional QWERTY keyboard and physical mouse setup in terms of overall usability. Compared to the QWERTY system, the Tap Strap 2 falls short in the key areas of learnability, efficiency, and accuracy. The two main niches this device would seem to satisfy would be VR users or the visually impaired. However, the device's VR capabilities did not work properly in this experiment, exposing them as error-prone at best, while it is surpassed by systems such as BrailleTouch for blind users in terms of learnability, since it does not require the user to learn a new alphabet. This is not necessarily claiming that the Tap Strap 2 does not have the potential to become a force in HCI in the future, as it does have the advantage of portability, especially when used with mobile devices. However, the device currently has little practical use and needs improvement in learnability, accuracy, and efficiency to compete with the QWERTY system. The next step in this research will be to implement the RNN-based spell checker into the Tap Strap 2 and determine how much this model improves the accuracy of the Tap Strap 2's output.

References

[1] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2573–2620, 2017.

[2] L. Meli, D. Barcelli, T. L. Baldi, and D. Prattichizzo, "Hand in air tapping: A wearable input technology to type wireless," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 936–941, 2017.

[3] J. Taylor and K. Curran, "Glove-based technology in hand rehabilitation," in *Gamification: Concepts, Methodologies, Tools, and Applications*, pp. 983–1002, IGI Global, 2015.

[4] L. Jing, Z. Cheng, Y. Zhou, J. Wang, and T. Huang, "Magic ring: A self-contained gesture input device on finger," in *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, pp. 1–4, 2013.

[5] C. Amma, M. Georgi, and T. Schultz, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors," in *2012 16th International Symposium on Wearable Computers*, pp. 52–59, IEEE, 2012.

[6] C. Southern, J. Clawson, B. Frey, G. Abowd, and M. Romero, "An evaluation of brailletouch: mobile touchscreen text entry for the visually impaired," in *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 317–326, 2012.

[7] S. Mascetti, C. Bernareggi, and M. Belotti, "Typeinbraille: Quick eyes-free typing on smartphones," in *Computers Helping People with Special Needs* (K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, eds.), (Berlin, Heidelberg), pp. 615–622, Springer Berlin Heidelberg, 2012.

[8] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, "Typingring: A wearable ring platform for text input," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 227–239, 2015.

[9] M. Gordon, T. Ouyang, and S. Zhai, "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*, pp. 3817–3821, 2016.

[10] C. Bisdikian, "An overview of the bluetooth wireless technology," *IEEE Communications magazine*, vol. 39, no. 12, pp. 86–94, 2001.

[11] L. Bieber and M. Willemin, "Motion detection mechanism for laser illuminated optical mouse sensor," Jan. 25 2011. US Patent 7,876,307.

[12] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[13] H. Li, Y. Wang, X. Liu, Z. Sheng, and S. Wei, "Spelling error correction using a nested RNN model and pseudo training data," *CoRR*, vol. abs/1811.00238, 2018.

[14] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[15] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, PMLR, 2013.