An Explicable Keystroke Recognition Algorithm for Customizable Ring-Type Keyboards

Sha, Xiaopeng; Lian, Chao; Zhao, Yuliang; Yu, Jianing; Wang, Shuyu; Li, Wen Jung

Published in:
IEEE Access

Published: 01/01/2020

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2020.2968495

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher’s copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 09/11/2022
An Explicable Keystroke Recognition Algorithm for Customizable Ring-Type Keyboards

XIAOPENG SHA, (Member, IEEE), CHAO LIAN, YULIANG ZHAO, (Member, IEEE), JIANING YU, SHUYU WANG, AND WEN JUNG LI, (Fellow, IEEE)

1School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
2Department of Mechanical Engineering, City University of Hong Kong, Hong Kong

Corresponding author: Yuliang Zhao (zhaoyuliang@neuq.edu.cn) and Wen Jung Li (wenjli@cityu.edu.hk)

This work was supported in part by the National Natural Science Foundation of China under Project 61503322 and Project 61873307, in part by the Fundamental Research Funds for the Central Universities under Grant N172304035, in part by the Qinhuangdao Science and Technology Planning Project under Grant 201901B013, in part by the Shenzhen Science and Technology Innovation Commission under Grant JCYJ20150828104330541, and in part by the Hong Kong Research Grants Council under Project CityU 11204918.

ABSTRACT In our previous work, we developed an IMU (Inertial Measurement Unit) based smart ring that allows users to type characters without a physical keyboard and adopt well-known pattern recognition algorithms, such as Support Vector Machine (SVM), and Naive Bayes (NB), for keystroke recognition. However, these algorithms always require intensive computing resources or offer limited recognition accuracy. Moreover, they are often seen as black boxes incapable of providing readily comprehensible and visible clues for classification. This hampers the improvement of keystroke recognition accuracy and the ring-type virtual keyboard’s character layout design. Here we present a novel algorithm to recognize keystrokes in a fast and accurate manner. Firstly, the standard feature vector, including five attitude angle features and one acceleration feature, is built to express a specific stroke. Then, the feature vector of the testing keystroke is compared with the standard features. The most similar keystroke is matched and recognized after three times of voting. Based on this algorithm, we can identify the easily confused keystrokes and understand the mechanisms behind it. With this interpretability, we will be able to achieve the customized ring-type virtual keyboard application if necessary. The performance of this algorithm was evaluated by using a dataset with 1500 keystrokes of three different subjects. The results show that our algorithm is more effective in keystroke recognition than traditional algorithms for this ring-type keyboard. In addition to its application on virtual keyboards, this algorithm can also be potentially applied on other classification tasks with easy-to-understand results.

INDEX TERMS Virtual keyboard, feature analysis, keystroke recognition, motion detection, human–computer interaction, recognition algorithm.

I. INTRODUCTION

The virtual keyboard provides an effective way to address the limitations of physical keyboards, such as privacy leaks, poor portability, and typing noise [1]. Researchers have proposed many constructive schemes that have led to great advances in virtual keyboard applications [2]–[9]. One example is our previous work on a novel virtual keyboard. It is a smart ring with an embedded IMU [10], [11], which allows users to type on the table or in the air without a physical keyboard.

By utilizing this smart ring, the motion data (acceleration and angular velocity) of typing finger can be captured and transferred to the computing platform for keystroke recognition. The keystrokes are recognized by well-known pattern recognition algorithms such as Support Vector Machine (SVM) and Naive Bayes (NB). These algorithms were proven to work well in recognizing keystrokes when using a computer with an Intel (R) Core (TM) i3-8100 CPU and 8 GB of main memory (RAM).

However, these algorithms could not meet two ultimate goals of the smart ring based virtual keyboard. One is that the smart ring should also be applicable in mobile scenarios,
i.e., situations when a powerful computing platform is not available to run the keystroke recognition process. That is, ideally the smart ring should be able to transfer the recognized characters, instead of the raw data, directly to the display device. The other goal is that we should be able to customize the layout of virtual keyboard based on the users’ specific typing habits. Therefore, the feature distribution of each keystroke should be well understood firstly.

To solve these problems, two improvements should be made: 1) less computational requirement for lower power consumption and faster recognition speed; 2) higher recognition accuracy and interpretability. These two problems hinder the application of the virtual keyboards, so it is critical to develop innovative approaches to address them. However, most popular pattern recognition algorithms today, such as k-Nearest Neighbor (k-NN), Naive Bayes (NB), Decision Tree (CART, ID3, C4.5 and etc.), Support Vector Machine (SVM), and Artificial Neural Network (ANN) [23], [26] cannot meet these two requirements simultaneously. If we pursue a simple and clean algorithm, its accuracy cannot always meet the application requirements.

Our previous research [10] demonstrated that the k-NN algorithm can offer good recognition accuracy without the requirement of model establishment, but it requires a large storage footprint and a prolonged processing time [14]–[16]. SVM and ANN are very accurate algorithms, however, they often require considerable amounts of training time to construct the models [15], [17]. In general, algorithms with an optimization process usually involve too many variables, complex objective functions, and high computational complexity [18], [19]. The storage requirements and processing time make it difficult to achieve real-time and fast keystroke recognition [20], [21]. Additionally, some of these algorithms must rely on a big size of training samples to get a good classification model. This in turn requires additional computing resources, including memory space and computing consumption.

Decision Tree and NB consume relatively fewer computing resources, but their recognition accuracy and stability are limited. For most Decision Tree algorithms [22], [23] such as information gain based ID3, information gain rate based C4.5, and Gini coefficient based CART [24], a small difference in the datasets may lead to a large change in the structure of the optimal decision tree, and thus result in unstable recognition performance. On the other hand, NB assumes that each feature is independent from one another [24], [25]. It works in a simpler and faster way than the induction of decision trees, but its accuracy in recognizing keystrokes is not always satisfactory.

Inspired by NB and Decision Tree algorithms, we proposed an explicable keystroke recognition algorithm for the smart ring based virtual keyboard. Similar to decision tree algorithms [23], [26], our algorithm has the benefits of being explicable and time efficiency. The same as NB algorithm, it only considers each single feature dimension separately, revealing the probability distribution of each feature in partial. With this algorithm, we can achieve higher accuracy and faster speed as well as easy-to-understand recognition results.

In the study, we evaluated the algorithm’s performance with three subjects, and obtained high accuracy and stability of recognition results from all the subjects. Then, we compared our algorithm with other existing recognition algorithms in terms of recognition accuracy and running speed. Our algorithm was demonstrated to be more suitable for recognizing keystrokes of ring-type keyboards. Subsequently, we conducted a detail analysis on the characteristics of the algorithm. Finally, we demonstrated the effectiveness and feasibility of our virtual keyboard scheme with the new algorithm. The results show that this new methodology can help to improve computational efficiency and recognition accuracy.

II. EXPERIMENTAL SETUP
Our recognition method is based on the changes in attitude angle during typing. We found that the differently positioned keys on the keyboard layout result in a unique typing path of each keystroke, which can lead to different finger motions and gestures. This unique pattern of finger motion can be well described and quantified by the attitude angle.

A. HARDWARE MODEL
For angle estimation, the motion information, including acceleration and gyroscope output, was collected by an intelligent sensor embedded in the ring model, as shown in Fig. 1. The sensor board incorporates a microprocessor along with a Bluetooth wireless communication module (NRF52832 SoC), an integrated environmental sensing module (BME280), and a MEMS motion sensing module (MPU9250), which consists of a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer for motion data collection. The integration of Microprogrammed Control Unit (MCU) and Bluetooth (BLE) allows the SoC NRF52832 module to reduce the power consumption and circuit size to a level that meets the design requirements for the ring [27], [28]. The entire board can be operated with a 48mAh coin cell battery and a switcher. The intelligent ring has a size of 19mm × 17mm × 25mm and a total weight of 7.8 grams. Because of its small size and light weight, this small ring is easy to use compared to other virtual keyboards. It can also be used in applications such as augmented reality, indoor navigation, and motion capturing.

B. EXPERIMENTAL METHOD
The keystroke recognition process entailed data collection, data preprocessing, angle estimation, motion segmentation, feature extraction, and data classification, as shown in Fig. 1. To avoid chaos while typing, a standard typing pattern is required to achieve the finger positioning in the QWERTY keyboard layout, as shown in Fig. 1. A finger-to-key mapping was established to prevent most horizontal typing errors, such as the accidentally touching of the horizontally adjacent letters [29]. These laid the groundwork for
subsequent processes. Before typing, user should place four of their left fingers (except the thumb) over keys ‘A’, ‘S’, ‘D’ and ‘F’. These keys are defined as the Home Row keys. To match the movement of fingers with keys, the first thing to do is to associate each finger to a group of keys [10], [11]. The recognition method requires that the user should type letters by moving and bending fingers after the typing started [7]. As indicated in [30], it was suggested to wear the ring on the middle phalanx of finger to better describe the finger movement.

### III. METHOD

#### A. DATA-PREPROCESSING AND FEATURE SELECTION

In our scheme, the finger motion data is collected by the sensor on the ring and sent to a mobile device or other devices through BLE when the keys are pressed. The typing motion data can also be sent to a remote server using the cloud technology in the future. The data preprocessing process includes zero-bias compensation, as well as noise removal which is intended to discard zero bias and smooth the waveforms [31], [32]. The zero-bias compensation method is described as $a(i) = a_r(i) - a_0$, where $a(i)$ is the final sampling sequence after zero bias compensation, $a_r(i)$ is the raw data, and $a_0$ is the average data of the sampling sequence. Then we applied a moving average filter to remove noise and smooth waveforms.

In the experiments, we found that the raw data of the acceleration and angular velocity are not stable and recognizable. On the contrary, the attitude angle feature has the advantage in expressing the uniqueness of each keystroke. In Fig. 2, the superiority of attitude angle is well illustrated by comparing with acceleration and gyroscope data. Therefore, the attitude angle was applied in this work through the angle complementary filter algorithm [33], which is described as the following equation:

$$\theta_{\text{Angle}} = \alpha \cdot (\theta_{\text{Angle}} + \omega_{\text{Gyro}} \cdot dt) + (1 - \alpha) \cdot a_{\text{Acc}}$$

where $\theta_{\text{Angle}}$ is the estimated angle, $a_{\text{Acc}}$ is the angle estimated using acceleration, and the section $(\theta_{\text{Angle}} + \omega_{\text{Gyro}} \cdot dt)$ is the angle calculated by fusing the estimated angle of last moment and angle increment. $\omega_{\text{Gyro}}$ is the angular velocity, and $\alpha (0 < \alpha < 1)$ is the weight coefficient.

Finally, we utilized the difference in gyroscope data to detect the starting and ending points of the keystroke motion based on its empirical amplitude and time length. The details of this method were well described in our previous work [10].
TABLE 1. The selected features: one feature in acceleration and five attitude angle features of each-axis.

<table>
<thead>
<tr>
<th>NO.</th>
<th>Symbol</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAX</td>
<td>( \text{MAX}(x_{k_1}, x_{k_2}) )</td>
<td>The maximum value in acceleration z-axis from ( k_1 ) to ( k_2 ) in time series</td>
</tr>
<tr>
<td>2</td>
<td>AVG</td>
<td>( \frac{1}{k_2-k_1} \sum_{k=k_1}^{k_2} x_k )</td>
<td>The average value of angle in time series from ( k_1 ) to ( k_2 )</td>
</tr>
<tr>
<td>3</td>
<td>SD</td>
<td>( \frac{1}{k_2-k_1} \sum_{k=k_1}^{k_2} (x_k - \bar{x})^2 )</td>
<td>The standard deviation of angle in time series from ( k_1 ) to ( k_2 )</td>
</tr>
<tr>
<td>4</td>
<td>MED</td>
<td>( \text{median}{ (k_1 + k_2) / 2 } )</td>
<td>The value of the middle point in time series from ( k_1 ) to ( k_2 )</td>
</tr>
<tr>
<td>5</td>
<td>MED</td>
<td>The central value which divide the sampling data of angle into two parts after the sort</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ENG</td>
<td>( E = \sum_{i=1}^{15} x_i )</td>
<td>The energy over the time in angle from ( k_1 ) to ( k_2 ) in time series</td>
</tr>
</tbody>
</table>

In addition to the attitude angle, we also used acceleration data to represent the uniqueness of each keystroke. We extracted six kinds of features as the feature vectors for classification, including average value, median, midpoint, standard deviation, energy from 3-axis attitude angles, and extreme value from acceleration z-axis, to ensure higher accuracy, as shown in Table 1.

Each selected feature contains different motion information. For example, the average and median values indicate the distribution level of waveforms from different angles, while the standard deviation reflects the dispersion degree of the data. These statistical characteristics are suitable for motion recognition and widely used in the machine learning algorithms [34], [35].

Therefore, the final features of each keystroke captured from one ring can be described as the vector \( X (a_1, a_2, \ldots a_{15}, g) \), including 15 attitude angles features and 1 acceleration feature on Z-axis as shown in Fig. 3.

B. RECOGNITION ALGORITHMS

The algorithm named SCM (consists of Spatial distribution, Central value distance, and Modification algorithm) proposed here is a multi-level decision algorithm entailing three times of voting. The algorithm is based on the distribution of each feature dimension separately. It includes two main sub-algorithms for keystroke recognition: the spatial distribution-based voting algorithm and the central value distance-based voting algorithm. The first sub-algorithm is used for primary classification, while the second is used for secondary classification. In the two sub-algorithms, the class with the highest vote is considered the predicted class.

Given the widely varying range of these statistical characteristics, we used z-score standardization [36] to keep the independent features at the same scale before keystroke recognition. The formula is shown as follows:

\[
X_{\text{scale}} = \frac{X - \bar{X}}{\sigma}
\]

1) THE FIRST SUB-ALGORITHM BASED ON SPATIAL DISTRIBUTION

The first sub-algorithm is developed based on the spatial distribution of these features. For illustration, two features...
were randomly selected from the training sample database, and we show the distribution of the two features in Fig. 3. It can be observed that the distribution varies among different keystrokes, but it is not easy to recognize keystrokes with one or more cut-off rules directly. Based on this observation, we analyzed the features further by adding some new restrictions. As shown in Fig. 3, there are two types of relationships (intersection type and independent type) between the features for different keys. Take keys ‘E’ and ‘F’ in ‘Feature A’ as an example (intersection type), there is an intersection in the distribution of their features, which cannot be separated completely. With the other type (independent type), as shown in ‘Feature B’, the two features are independent from each other and therefore can be separated completely. Consequently, we designed a voting algorithm based on the spatial distribution of different features. Finally, we constructed the upper and lower thresholds, which are similar to the bounds of the interval, for each feature to fit the data distribution.

Then, we set the upper and lower thresholds for each feature automatically against the distribution range of the template. According to the experimental results, the thresholds \( h_l \) and \( h_u \) are not strictly equal to the bounds of the interval. There is always a buffer zone. As shown in Fig. 3, it is described as outliers in the template dataset. The outliers have bad impacts on the setting of parameters. Here, we applied Rajda criterion \([37, 38]\) to avoid such impacts. The criterion provides a statistical basis to estimate the outliers, which can be described by the following mathematical model:

\[
P(\mu - 3\sigma < X < \mu + 3\sigma) \approx 0.9973
\]

where \( \mu \) represents the average value, and \( \sigma \) represents the standard deviation. The buffer zone (3\( \sigma \)) can also be adjusted as necessary. The function of this value can also be seen as the slack variable to increase fault tolerance of the threshold and avoid over fitting.

According to the algorithm, for the testing vector \( X(x_1, x_2, \ldots, x_n) \) of attitude angle, we constructed the thresholds \( 2 \times n \) matrix \( T \) for each feature element, which is described as follows:

\[
T = \begin{bmatrix}
  h_l(1) & h_l(2) & \cdots & h_l(n) \\
  h_u(1) & h_u(2) & \cdots & h_u(n)
\end{bmatrix}
\]

The column represents the different features of a certain keystroke. If any element \( x_i \) in the vector \( X \) satisfies the condition,

\[
f(x_i) = \begin{cases} 
  1 & \xi h_l(i) < x_i < \xi h_u(i) \\
  0 & \text{else} 
\end{cases} \quad i = 1, 2, \ldots, n
\]

\[
V_j = \left( \sum_{i=1}^{n} f(x_i) \right), \quad j = 1, 2, 3 \ldots k
\]

\[
\text{Class} = \arg \max_j V_j
\]

where \( k \) is the number of keys. Then according to the votes \( V_j \), it will be divided into the corresponding classes. As shown in ‘Feature A’, we allow it vote to different classes simultaneously because of the cross distribution of the feature template. Finally, the vector \( X \) will be classified into the class that has the most votes.

Analysis of the features shows that keys can be easily recognized if they are far away from each other in the keyboard layout, such as ‘T’ and ‘X’, because there are more independent type features that are easy to distinguish. However, when the same method is applied to some of the adjacent keys, it cannot work well sometimes. This is because the keys have similar features, and the votes in features between them are insufficient to support keystroke recognition. We conclude that the closer the keys are to each other in the horizontal line, the more similar their features are. This means independent type features of different keystrokes are deeply overlapped with each other. Despite its unfitness for recognizing some of the adjacent keys, this voting method can perform most of the recognition work, as evidenced by the experimental results.

2) THE SECOND SUB-ALGORITHM BASED ON CENTRAL VALUE DISTANCE

The algorithm described in part 1) shows that, most of the testing vectors can be divided into a particular class. However, some testing vectors have the same votes and therefore cannot be classified. The finger motions when pressing keys ‘W’ and ‘E’, for example, are similar, and therefore may cause the recognition to fail. This requires a second classification to improve the algorithm. Considering the distance-based method is used in most machine learning algorithms for pattern recognition, we designed another voting algorithm based on central value distance. Here is how it works.

The main idea of this algorithm is to use the central value of the template to quantify the distance between training matrix and testing vectors. The median value was selected as the central value, which is proven to be a better choice than the average value. The median matrix \( C \) can be expressed as follows:

\[
C = \begin{bmatrix}
  c_1(1) & c_1(2) & \cdots & c_1(i) & \cdots & c_1(n) \\
  c_2(1) & c_2(2) & \cdots & c_2(i) & \cdots & c_2(n) \\
  \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
  c_j(1) & c_j(2) & \cdots & c_j(i) & \cdots & c_j(n) \\
  \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
  c_p(1) & c_p(2) & \cdots & c_p(i) & \cdots & c_p(n)
\end{bmatrix}
\]

where \( n \) is the number of the features. Then, we calculated the distance between each testing vector \( X(x_1, x_2, \ldots, x_n) \) and the central value vector \( C_j \) of the templates. Where \( C_j \) represents the template of \( j^{th} \) character (\( C_j \in C \)). Then the class with the highest number of votes will be identified by the following formula:

\[
g(x_i) = \begin{cases} 
  1 & |x_i - c_i| > \forall |x_j - c_i| \\
  0 & \text{else} 
\end{cases} \quad i = 1, 2, \ldots, n, \; i \neq j
\]
\[ V_j = \left( \sum_{i} g(x_i) \right), \quad j = 1, 2, 3, \ldots, p \quad (10) \]
\[ \text{Class} = \arg \max_{j} V_j \quad (11) \]

Then the distance vector defined as \( |x_i - c_j| \) will be constructed. The smallest distance will be voted and the final class will be identified by the total number of votes. Equation (9) obtains the votes of different classes. And, equation (10) returns the label of class with the most votes.

The first sub-algorithm is designed for rough estimation of the most possible keystrokes. Furtherly, the second sub-algorithm finds the best matching keystroke in the possible ones. These two sub-algorithms complement with each other. None is dispensable for an ideal recognition performance.

3) THE THIRD SUB-ALGORITHM BASED ON MODIFICATION RULE

The two algorithms above can work well for recognition most of keystrokes. However, they may fail to recognize keystrokes that are typed with very little finger motion, and this often occurs in the Home Row keys with small changes of attitude angle.

The solution to this problem is the maximum features of acceleration, as calculated in Table 1. When pressing the Home Row keys, the acceleration z-axis of the sensor worn on the touching finger exhibits a more obvious change than that of those on the non-touching ones, as shown in Fig. 5. Thus, we used the acceleration feature for further keystroke recognition. The Home Row keys were easily selected by the two voting algorithms above because of little changes in attitude angle when pressing them. Then the sensor with maximum value \( y_{\text{max}} \) in acceleration can be detected and mapped into keys. With this principle, we can easily identify and exclude the typing finger and recognize the keys pressed according to the typing standards.

IV. RESULTS AND ANALYSIS

In the study, we collected the dataset from three participants. They were allowed to type in their own typing speed and gesture during the experiment. The 15 alphabetic keys typed by left-hand fingers were sorted randomly. Then, 100 sets of typing motion data of each key were collected and translated into features for establishing the mapping relation with keystrokes. In the experiment, 70% data of the feature was randomly applied as templates to identify. The remaining data was used for testing to ensure the training and test sets were separated.

A. RECOGNITION RESULTS

1) RECOGNITION RESULTS WITH FOUR WEARABLE SENSORS

In the experiment, we obtained the recognition results of the three subjects with 70 training samples, as shown in Fig. 6. The average recognition accuracy is 99.36%, 99.08% and 97.98%, respectively, for the three subjects with four wearable sensors. The color map suggests some common recognition errors among the subjects. For instance, key pairs ‘W’ and ‘E’ and ‘C’ and ‘X’ are often easily confused and where errors mostly occur. A common challenge facing today’s virtual keyboards is that adjacent keys in the same row or column are often recognized incorrectly. However, the accuracy of SCM algorithm can remain high in spite of these differences.

2) RECOGNITION RESULTS WITH THREE AND TWO WEARABLE SENSORS

For convenience, we reduced the number of the wearable sensors worn on the fingers based on their relationships during typing. In the experiment, we found that the non-touching fingers move along with the touching finger during the typing process, which related to the typing habits. Some researchers
have conducted studies on the relationship between touching and non-touching fingers when pressing keys and gotten certain achievements [29].

Thus, it is possible for us to recognize keystrokes with less sensors on. As shown in Fig. 3, for the keys ‘E’ and ‘F’, they can be recognized by the only two listed features extracted from the sensors worn on middle and index fingers. This phenomenon tells us that even though there may be no feature extracted from the corresponding typing finger, it can also be accurately recognized form the limited features from other non-touching fingers. And it gives us enlightenment that we can make further recognition with reduced sensors.

As described in the third sub-algorithm of SCM, we compared the acceleration data to identify the working finger. Fewer rings means less acceleration z-axis data available from sensors on fingers to modify the algorithms. This will in turn lead to a weaker influence on the keystroke recognition of Home Row keys, especially when the number of sensors is reduced from four to two or one. The recognition results with three and two wearable sensors are shown in Table 2.

We can achieve a high level of accuracy with the algorithms. Commonly the recognition accuracy declines as the number of sensors reduces but stays at a relatively stable level. To sum up, our approach is effective and reliable to recognize keystrokes. A higher accuracy is observed from all the three subjects when sensors are worn on the little and middle fingers. Hence, we provide a detailed description of the recognition results when sensors are worn on these two fingers for each subject, as shown in Fig. 7.

The accuracy is high for both all the keys in general and for each key individually. However, the accuracy is relatively low for some keys, such as ‘S’ and ‘W’. The difference in accuracy of misclassified keys among different subjects is also magnified, which is much more obvious in keys ‘S’ and ‘W’, for example.

3) RECOGNITION RESULTS WITH ONE WEARABLE SENSOR

We also explored the difference in accuracy with one sensor worn on left-hand fingers. To get a higher accuracy, we once again modified the algorithm, which is shown as Fig. 8.

In the algorithm, we compare the maximum data of acceleration data with the thresholds $d_1$ and $d_2$. $d_1$ is the threshold for acceptation and $d_2$ is the threshold for rejection. If necessary, the two thresholds can be tuned for a given individual to enable dramatically increased accuracy. The results are shown in Table 3.

As the recognition result shows, there is a drop in the recognition accuracy when the number of sensors is reduced to one. However, with SCM algorithm, the average recognition accuracy can still keep in a relatively high level with the proportion of 90%. Even better, the recognition rate of Subject B can achieve 95.51%, which fully demonstrates the superiority of the algorithm.

B. INTERPRETABILITY OF SCM ALGORITHM

The excellent interpretability of SCM algorithm brings it one remarkable advantage. Through the algorithm, we can learn
TABLE 3. Mean recognition accuracy with one wearable sensor worn on fingers (%). (Notations: 1-little finger, 2-ring finger, 3-middle finger, 4-index finger).

<table>
<thead>
<tr>
<th>Subject</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>89.13</td>
<td>92.16</td>
<td>91.78</td>
<td>87.16</td>
</tr>
<tr>
<td>B</td>
<td>95.51</td>
<td>93.62</td>
<td>93.47</td>
<td>86.76</td>
</tr>
<tr>
<td>C</td>
<td>86.11</td>
<td>79.69</td>
<td>86.07</td>
<td>77.89</td>
</tr>
<tr>
<td>Average</td>
<td>90.25</td>
<td>88.49</td>
<td>90.44</td>
<td>87.27</td>
</tr>
</tbody>
</table>

FIGURE 9. Confusion matrix of subject C with one sensor on the left middle finger for a recognition trial.

FIGURE 10. Distribution of features for keys ‘W’ and ‘E’ and the missed testing vectors of key ‘W’. (a) Envelope line of template ‘W’; (b) envelope line of template ‘E’; (c) envelope line of misclassified testing vectors of ‘W’; (d) mean values of three types of data respectively. In figure (d), the gray line represents key ‘E’, the red line represents key ‘W’, and the blue dotted line represents the misclassified testing vectors of key ‘W’.

The distribution difference of the typing characteristics in different keystrokes and explore the classification mechanism of different keystrokes.

In this part, we will discuss the interpretability of SCM from three aspects: 1) main source of misclassification, 2) good performance with small sample size and 3) potentially customized application for each individual.

1) MAIN SOURCE OF MISCLASSIFICATION
The deep understanding of the causes of misclassification through the algorithm is a crucial way to improve the algorithm on efficiency, accuracy and stability. In the experiment, we show the recognition confusion matrix of subject C with one sensor worn on the middle finger, as shown in Fig. 9.

The recognition accuracy rate is much lower for the keys ‘A’ and ‘W’, while it is higher for the keys ‘C’ and ‘V’. But this recognition results are dramatically different from what we obtained by having two, three or four sensors worn on the fingers. Therefore, we make the error recognition analysis specifically.

For effective solution to the low accuracy of these missed keys, we extracted the missed testing vectors of key ‘W’ for analysis. Considering the recognition algorithms, we drew the distribution curves for each sample, which are shown in Fig. 10.

From Fig. 10, we see that the testing vectors are more similar to the misclassified class. We applied the widely used metric Pearson correlation coefficient [39] $\rho$ to quantify and evaluate the similarity. Given two variables $X$ and $Y$, the correlation coefficient can be measured by the following equation:

$$\rho_{XY} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$  \hspace{1cm} (12)

where $\text{cov}$ represents the covariance operator, $\sigma$ the standard deviation, $E$ the expectation, and $\mu$ the mean value.

The correlation coefficient between templates ‘W’ and ‘E’ and that between the missed testing vectors of ‘W’ are 0.9738 and 0.9846, respectively. The correlation coefficient between the two templates is 0.9786. We can conclude that there is little difference between ‘W’ and ‘E’, and the correlation between the missed testing vectors of template ‘W’ is weaker than that of ‘E’. This demonstrates that the sensor worn on only one finger in each hand cannot provide sufficient motion information for recognition because of the weak correlation among fingers when typing some keys.

Thanks to the interpretability of SCM algorithm, we can make a customized application for each individual. Since SCM algorithm is synthetically classified according to the vote of each feature, the appropriate selection of features may be beneficial for further improvement of the recognition rate. For example, adding new features or eliminating interference features are both the good choice.

2) GOOD PERFORMANCE WITH SMALL SAMPLE SIZE
Another advantage of SCM algorithm is that even small sample size it can still have the ability to describe the real distribution of features, and obtain good recognition performance. With this characteristic, accurate keystroke recognition can still be performed even though there are not enough training samples. This characteristic can be reflected through the recognition accuracy of data sets in different size.
Therefore, we conducted the experiments and applied different scale of samples for illustration. We increased the number of training samples per key from 30 to 70. The recognition experiment was repeated 10 times for each training sample size on the three subjects, the results are shown in Fig. 11.

It is obviously that SCM algorithm achieves a high accuracy even at small training samples size. It means that even with a small sample size we can still accurately estimate the overall feature distribution for different keys. Fig. 11 reveals that there is an upward trend in the recognition accuracy with an increase in the training sample size. As the sample size increases, the recognition accuracy of subjects can improve slowly until reaching the saturation value. And Fig. 11(a)~(c) also reveals that the training samples size impact on recognition accuracy varies from one subject to another, which can be easily concluded in the upward trend of three subjects.

3) POTENTIALLY CUSTOMIZED APPLICATION FOR EACH INDIVIDUAL

Another manifestation of interpretability is that we can see the diversity in the typing behaviors of different individuals through SCM algorithms. Through this characteristic, we can be able to achieve the customized application for each individual specially.

The diversity exists among different individuals, although the keys are uniquely positioned in the keyboard layout. Actually, the feature distribution among different subjects then exists some minute difference. The keystroke waveforms of a single key for different individuals are often distorted and in departure with each other due to the different formations and directions of the finger motion when pressing that key. The departure in this case is much bigger than when it comes to the keystrokes of the same key for one particular individual. We can infer that this difference may partly cause differences in the keystroke recognition results. Besides, the typing proficiency can cause the diversity of different subjects based on the experiment observation, which can also lead to the difference in recognition accuracy. This pattern can be defined as diversity in typing behaviors of individuals. As shown in Fig. 6, a phenomenon is that keys ‘E’ and ‘W’, for instance, can easily be confused by subject B due to their similar typing behaviors, but they are hardly confused by subject C who tends to confuse keys ‘C’ and ‘X’.

As analysis above, with SCM algorithm we can learn the feature distribution of different keys easily. In this way, we can have a deep understanding of typing pattern of different users, which can make us not only be able to adjust the parameters in the algorithms but select the suitable recognition features for each individual. Besides, we will also be able to adopt other method to increase the difference of different keys for each individual, such as keyboard layout design and etc.

C. COMPARISON IN RECOGNITION PERFORMANCE

We evaluated the performance of the algorithms based on recognition accuracy and running speed. The evaluation was mainly through the comparison among universal pattern recognition algorithms, such as decision tree (CART), Naive Bayes (NB), k-Nearest Neighbor (k-NN), Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM).

1) COMPARISON IN RECOGNITION ACCURACY

To better evaluate the effectiveness of the algorithms, we conducted the experiment and summed up the recognition accuracy. The recognition accuracy of three subjects is shown in Fig. 12.

We can see that SCM algorithm performs more effectively than some other algorithms. It can achieve the recognition accuracy about 99% for each subject, which impressed us a lot. Specifically, the CART, NB, and k-NN machine learning algorithms show a lower and less stable level of accuracy for three subjects than the LDA, SVM and the SCM algorithm we proposed. This result shows that the SCM algorithm is comparable and even better than some of the recognition algorithms in recognizing keystrokes. Admittedly, this algorithm offers a relatively lower level of accuracy and stability than LDA and SVM, but the gap is negligible. Particularly, when subject C shown a low proficiency in typing, the recognition accuracy dropped comparing with that of subjects A and B.
FIGURE 13. Comparison between different recognition algorithm in terms of recognition accuracy and running speed.

2) COMPARISON IN RUNNING SPEED
We used the predicting time of 30 testing vectors to reflect the running speed. To better demonstrate the superiority of the algorithm, we added recognition accuracy into the figure for comparison, as shown in Fig. 13.

We can easily conclude that SCM shows the best performance as expected. The SVM and LDA algorithms perform better in recognition accuracy, but much worse recognition speed. As revealed by the recognition speed, the high accuracy of these two algorithms is at the cost of a prolonged recognition process. The training model and parameters can be established better by increasing the sample size, which has no impact on the computational complexity, because the model size will remain unchanged.

The evaluation of recognition accuracy and running speed shows that the SCM algorithm has an excellent performance than traditional algorithms such as SVM and CART. It is noteworthy that SCM is promising for further application due to its distinctive advantage in speed and memory consumption.

D. DISCUSSION
The characteristics of SCM algorithm can be summarized as follows. We conclude that even with small size of training samples, it can still achieve a high recognition accuracy. And it is very easy to establish such a model for a particular subject. Once the training sample is given, the model size, including upper and lower thresholds, will be determined and remain unchanged despite changes in the sample size. The significant advantages of the algorithm are simplicity, yet effectiveness, and computation efficiency. This consequently ensures less memory consumption and faster testing speeds. The experimental result also shows that SCM algorithm can achieve high keystroke recognition accuracy. As a result, it is more suitable for this ring-type virtual keyboard compared with other machine learning algorithms. It can also be used in many operating environments in which many universal recognition algorithms cannot work well.

Another great advantage of SCM algorithm is interpretability, which can lead to the customized virtual keyboard applications. It potentially includes the customized parameters, the customized features of the algorithm, and the customized keyboard layout. With this characteristic, we can learn the real feature distribution of keystrokes for each key. Specially, the algorithm allows us to better learn the specific typing pattern of each individual and help design new keyboard layouts. This means that this algorithm can guide us through customizing keyboard layouts for different individuals to their specific typing habits. Besides, distinguishing difference among the easily confused keys through the algorithm can lead to a keyboard layout that is better suitable for this virtual keyboard scheme and reach improved accuracy and comfort. Further improvements will be possible by adjusting the key size, the key position, the layout, and even the shape [40]–[42].

One thing needs to be noticed is that this ring-type virtual keyboards emphasizes the personality of each individual. The participants’ typing patterns distinct from each other, but they stay consistent over the time. Some researchers have studied this personalized typing pattern on the touchscreen virtual keyboard [40], [41]. Similarly, this typing pattern also occur and has a profound influence on our recognition process. In this case, the recognition accuracy varies significantly among different individuals.

V. CONCLUSION
Here we proposed an explicable keystroke recognition algorithm for customizing a ring-type virtual keyboard. Based on the three times of voting for decision, the algorithm is developed against the distribution of each feature, which combines the characteristics of decision tree and Native Bayes algorithms. This algorithm outperforms their counterparts in both accuracy and running speed. Therefore, it can potentially be used in other environment, especially hardware systems with low memory capacity and weak computing power. More importantly, the algorithm is designed based on the real feature distribution of each class, allowing us to easily understand the difference between classes. In this paper, we focused on the interpretability of the algorithm, which provide the possibility to delve the customized application for this ring-type virtual keyboard. Therefore, we will be able to develop better-performing virtual keyboards for special applications. The use of this new algorithm can also be extended into many other pattern recognition applications such as motion recognition and object classification.

ACKNOWLEDGMENT
(Xiaopeng Sha and Chao Lian contributed equally to this work.)

REFERENCES


CHAO LIAN received the B.E. degree from Liaoning Technical University, in 2017. He is currently pursuing the M.S. degree with the School of Control Engineering, Northeastern University at Qinhuangdao, China. His current research interests are in the areas of wearable cyber physical devices, inertial measurement unit, motion analysis, machine learning, and artificial intelligence.

YULIANG ZHAO (Member, IEEE) received the B.S. degree in mechanical engineering from the Hubei University of Automotive Technology, the M.S. degree in mechanical engineering from Northeastern University, and the Ph.D. degree in mechanical and biomedical engineering from the City University of Hong Kong, in 2016. He is currently an Assistant Professor with Northeastern University at Qinhuangdao, Qinhuangdao, China. His research interests include intelligent sensors, machine learning, motion analytics, and big data analyses. His recent work involves applying these technologies to single-cell and biomechanical analyses.

SHUYU WANG received the B.S. degree in mechanical engineering from the Huazhong University of Science and Technology, in 2013, and the Ph.D. degree in mechanical engineering from Stony Brook University (SUNY), in 2017. He has semiconductor industry experience with GlobalFoundries. He is currently an Associate Professor with the Control Engineering Department, Northeastern University at Qinhuangdao. His current research interest includes biosensing and machine learning’s applications in sensing.

JIANING YU is currently pursuing the bachelor’s degree with Northeastern University at Qinhuangdao, China. His research interests focus on embedded systems development, wearable network physical devices, inertial measurement units, and artificial intelligence.

WEN JUNG LI (Fellow, IEEE) received the B.S. and M.S. degrees in aerospace engineering from the University of Southern California (USC), in 1987 and 1989, respectively, and the Ph.D. degree in aerospace engineering from the University of California at Los Angeles (UCLA), in 1997. From September 1997 to October 2011, he was with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. His industrial experience includes Aerospace Corporation, El Segundo, CA, USA, the NASA Jet Propulsion Laboratory, Pasadena, CA, USA, and Silicon Microstructures, Inc., Fremont, CA, USA. He is currently a Chair Professor with the Department of Mechanical Engineering and an Associate Provost (Institutional Initiatives) with the City University of Hong Kong. His current research interests include intelligent cyber-physical sensors, super-resolution microscopy, and nanoscale sensing and manipulation. He served as the President for the IEEE Nanotechnology Council, from 2016 to 2017.