Research Article

Feature Extraction of Sequence of Keystrokes in Fixed Text Using the Multivariate Hawkes Process

Chang Zhang , Yuchen Zhang , and Fulin Li

Information Engineering University, Zhengzhou, Henan, China

Correspondence should be addressed to Chang Zhang; zhang_chang_xd@163.com

Received 21 December 2020; Accepted 13 May 2021; Published 24 May 2021

Academic Editor: Gen Q. Xu

Copyright © 2021 Chang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a new method of extracting the features of keystrokes. The Hawkes process based on exponential excitation kernel was used to model the sequence of keystrokes in fixed text, and the intensity function vector and adjacency matrix of the model obtained through training were regarded as the characteristics of the keystrokes. A visual analysis was carried out on the CMU keystroke raw data and the feature data extracted using the proposed method. We used one-class classifier to compare the classification effect of CMU keystroke raw data and the feature data extracted by the Hawkes process model and POHMM model. The experimental results show that the feature data extracted using the proposed method contains rich information to distinguish users. In addition, the feature data extracted using the proposed method has a slightly better classification performance than the original CMU keystroke data for some users who are not easy to distinguish.

1. Introduction

The keyboard is a common man-machine interface in information systems. By recording the time when people press and release the keys (keystrokes) and extracting features from the keystrokes, we can apply the data extracted from the keystrokes for authentication, identity verification, and intrusion detection of the information system. Similar to studies on the voiceprint, studies on the analysis methods of keystrokes can be divided into two categories, namely, content-independent and content-related. In content-independent scenarios, the user types fixed text, e.g., username and password, when logging into the information system, which then recognizes or authenticates the user's identity by extracting the features of the keystrokes. In this case, for user identification, instead of considering the typed text, the time interval between pressing and releasing the keys is formed into a multidimensional feature vector (one dimension for one action) as the original feature according to the sequence of keystrokes in the fixed text. In content-related scenarios, the system identifies the user when he or she is using the information system. Under such circumstances, the user types arbitrary text (also known as free text). To eliminate

the influence of text content on the features of keystrokes in user identification, the average time spent by the user in typing common combinations of English letters, e.g., th, is, and ing, is transformed into a multidimensional feature vector (each combination of letters as one dimension) as the original feature.

In most previous studies, the above eigenvalues were taken as discrete quantities for modeling, and various classifiers were proposed by means of statistics, machine learning, and deep learning. However, the identification accuracy of these classifiers fails to meet practical requirements. One of the reasons is that the original features of the keystrokes or the features extracted based on the original features are insufficient to achieve highly accurate classification. Moreover, the dynamic change of keystrokes is another main cause. With the eigenvalues of keystrokes regarded as discrete quantities, it is hard to capture the dynamic changes of features. Hence, in this study, we established a model based on the sequence of keystrokes from the temporal perspective, which is more consistent with the dynamic change of keystrokes. Hidden Markov model (HMM) is a common temporal model for analyzing keystrokes. Its hidden state corresponds to keystrokes and

emission probability corresponds to the probability distribution of the time interval between keystrokes. This model fails to consider the continuity of keystrokes. In addition, as a generation model, although the trained HMM contains the characteristics of users' keystrokes, it cannot describe the characteristics of a single sample. Temporal point process is a mathematical tool to describe discrete events in continuous time domain. In this study, considering the continuous sequence of keystrokes, we used the multivariate Hawkes process, which is a special temporal point process, to model the sequence of keystrokes in fixed text in order to analyze and extract the characteristics of the sequence of keystrokes.

2. Research Background

The samples related to the sequence of keystrokes mainly include key value, down time, and up time. The difference in the down time of adjacent keys is referred to as DD time or digraph; the difference in the down time of two keys with one key in the middle is called trigraph; and the difference in the down time of two keys with $n-1$ keys in the middle is known as n -graph. The up time of one key minus the down time of this key equals the hold time. For adjacency keys, the down time of the latter key minus the up time of the former key equals the UD time. The data about the sequence of keystrokes in fixed text generally includes DD time, UD time, and hold time [1, 2]. Figure 1 shows the calculation of the time interval with the fixed text "GEN" as an example. It should be noted that Figure 1 shows only one scenario about the sequence of keystrokes.

In a series of studies by Kevin et al. [1–3], the sequence of keystrokes was directly input into the classifier as the feature vector. In some studies, data about the sequence of keystrokes was processed. Bergadano et al. [4] calculated the average trigraph of the text and ranked the combinations of letters per the average trigraph as the feature of keystrokes. Robinson et al. [5] regarded the mean and variance of hold time as the feature. Monroe and Rubin [6] took the average and variance of the time spent in typing common combinations of letters, e.g., *th* and *he*, as the characteristics of the keystrokes. In a study by Araújo et al. [7], the averages and variances of UD time, DD time, and DU time served as the characteristics of the keystrokes. In a study by Epp et al. [8], besides various time intervals, the number of errors, the number of keystrokes, and the number of characters were also used as the original features, from which the feature subset was selected by the feature selection approach. With some statistical values (e.g., average, equation, skewness, autocorrelation, and moment) and information measurements (e.g., entropy) of the time interval as the features, Ulinskas et al. [9] applied a feature selection approach to select a feature subset from them. Based on fuzzy logic, de ru and Eloff [10] divided the time interval into four categories, that is, very short, short, moderately short, and somewhat short, as the characteristics of keystrokes. Mondal and Bours [11] took the key values of adjacency keys as well as hold time, UD time, UU time (up time 2–up time 1), and DD time as the feature vectors. Apart from the time interval, the key pressing strength, the position of the key on the keyboard,

error frequency, and keystroke sound could also be used as features [12]. Lin et al. [13] input the key value, down time, UD time, and DD time into a convolution neural network as the original feature matrices. In the abovementioned studies, the discrete eigenvalues were combined to design the corresponding classifier and obtain the classification results. Sung and Cho [14] used genetic algorithm-based SVM wrapper ensemble approach to select feature, and some ensemble learning methods [15–17] can be applied to this field. These methods failed to effectively explain which features played a role in identifying users.

However, in some studies, the temporal characteristics of the sequence of keystrokes were also taken into consideration. Alpar [18] transferred the sequence of keystrokes in fixed text from the time domain to the frequency domain for analysis, which was not accurate enough due to the small number of keystrokes. HMM is a temporal model commonly used to analyze keystrokes [19–21]. Compared with the methods that take the sequence of keystrokes as a feature vector composed of the time interval between the actions of pressing and releasing the keystrokes, the temporal modeling method can make full use of the available data. The partially observable hidden Markov model (POHMM) [21] proposed by Monaco from the US Army Research Laboratory has an ideal performance in this aspect. The hidden variables of this model include two hidden states, i.e., positive and negative, based on which the sequence of observed keystrokes is generated. Corresponding to the state transition matrix and emission probability of the model, the features of keystrokes are the overall features of the training sample set rather than the features of a specific sample. Therefore, the features of a single sample cannot be obtained based on this model. In contrast, the temporal model proposed in this study is able to extract the features of a single sample for visual analysis.

3. Model Description

3.1. Description of Keystrokes. Keystrokes can be divided into two actions, i.e., pressing (down) and releasing (up). When the user types text, by recording the key value, action type, and time t , the information system's sequence of keystrokes can be represented as $\{k_1^{\text{down}}, k_1^{\text{up}}, k_2^{\text{down}}, \dots, k_N^{\text{up}}\}$, where the subscript i in k_i^{type} denotes the i -th key pressed; k_i corresponds to the key value (e.g., key "A"); type represents the type of action (e.g., "down"); and k_i^{type} stands for the time when the keystroke occurs. This study is based on the premise that the user types text without errors (no "backspace" key in the sequence of keystrokes). The key values of the sequence of keystrokes are the content of the text.

Figure 2 shows the sequence of keystrokes event for the fixed text "hello." The first case was generated by hitting the keyboard in the rhythm of press-release-press-release. Poking the keyboard with a single finger will produce such a sequence. However, people usually use both hands to strike keys, and the different fingers of the left and right hands strike different keys at the same time. The sequence of keystrokes in the second and third cases will appear as a result. We can record the time the finger presses the k_i key as

UD time 1/flight time = Down time 2 – Up time 1
 DD time 1/digraph 1 = Down time 2 – Down time 1
 Hold time 1/dwell time 1 = Up time 1 – Down time 1

Trigraph 1 = Down time 3 – Up time 1
 3 – graph 1 = Down time 4 – Up time 1
 4 – graph 1 = Down time 5 – Up time 1

FIGURE 1: Calculation of the time interval of keystrokes (with “gen” as the case).

k_i^{down} . Then, the finger will “bounce” to release the k_i key. Therefore, there is a triggering relationship between pressing and releasing; that is, k_i^{down} triggers k_i^{up} . Next, consider the relationship between k_i^{down} and k_{i+1}^{down} . Since typing determines the text, k_i^{down} must happen before k_{i+1}^{down} . That is, after the user types k_i , k_{i+1} will be typed immediately. Therefore, there is also a triggering relationship between them. Finally, consider the relationship among k_i^{up} , k_{i+1}^{down} , and k_{i+1}^{up} . Unlike the down action, k_i^{up} does not necessarily occur before k_{i+1}^{down} and k_{i+1}^{up} ; it may also occur after them. Therefore, there may or may not be a triggering relationship between k_i^{up} and k_{i+1}^{up} , k_{i+1}^{down} and k_{i+1}^{up} . In summary, the triggering relationships of the keystroke events of fixed text are the following:

- (1) $k_i^{\text{down}} \rightarrow k_{i+1}^{\text{down}}$, corresponding to the mutual triggering relationship $\omega_{i+1,i} > 0$
- (2) $k_i^{\text{down}} \rightarrow k_i^{\text{up}}$, corresponding to the self-triggering relationship $\omega_{i,i} > 0$

The sequence of keystroke $\{k_i\}_{i=1}^N$ contains N -type events; k_i^{down} and k_i^{up} are two actions belonging to the i -th event. Figure 3 depicts an example of a sample. The down and up actions of similar events are self-triggering, and there is a mutual triggering relationship between different types of events (the down action corresponding to the previous text triggers the down action corresponding to the following text: the trigger value between adjacency events is larger; the farther the event is, the smaller the trigger value is). There is no triggering relationship between other events (i.e., the trigger value is equal to 0). The triggering relationship can be expressed as the following matrix:

$$\mathbf{W} = \begin{bmatrix} \omega_{11} & 0 & 0 & 0 \\ \omega_{21} & \omega_{22} & 0 & 0 \\ \omega_{31} & \omega_{32} & \omega_{33} & 0 \\ \omega_{41} & \omega_{42} & \omega_{43} & \omega_{44} \end{bmatrix}. \quad (1)$$

3.2. Multivariate Hawkes Process. The multivariate Hawkes process [22] is a counting process corresponding to a sequence of events composed of multiple types of

(multidimensional) events. There is an incentive relationship between these events. “Multiple” corresponds to multiple types. There are two ways to define the multivariate Hawkes process—conditional intensity function and Poisson cluster process. Both methods have their own advantages. Conditional intensity functions can be superimposed and combined; their formula description is flexible and concise; and they are easy to calculate. The clustering Poisson process is suitable for deriving the first or second moment metrics. This article adopts the conditional intensity function definition method. Suppose that the multivariate counting process is $N(t) = \{N_i(t)\}_{i=1}^D$, where $N(0) = 0$; its dimension is D ; and the conditional intensity function is $\lambda(t) = \{\lambda_i(t)\}_{i=1}^D$. The specific form is

$$\lambda_i(t) := \mu_i(t) + \sum_{j=1}^D \int_{-\infty}^t g_{ij}(t-\tau) \cdot dN_j(\tau), \quad (2)$$

where μ_i is a constant; $\mu_i \in \mathbb{R}_+$, $\mu(t) = \{\mu_i(t)\}_{i=1}^D$. The D -dimensional intensity function vector represents the external part of the intensity function of the temporal point process i (the intensity triggered by an external event); $\mathbf{G}(t) = \{g_{ij}(t)\}_{i,j=1}^D$ is a $D \times D$ -dimensional excitation kernel matrix; and the excitation function $g_{ij} \geq 0$ describes the endogenous influence (incentive) of events that have occurred in the current j -th dimension of the multivariate Hawkes process on the intensity of the i -th dimension event. Formula (2) satisfies the following conditions:

- (1) $g_{ij}(t) \geq 0$
- (2) if $t < 0$, $g_{ij}(t) = 0$
- (3) $g_{ij}(t) \in L_1$ space

The integrable function $g_{ij}(t) \in \mathbb{R}_+$ is the element of the matrix $\mathbf{G}(t)$, called the excitation kernel function, which describes the incentive of j -type events to i -type events at time t during $t \in [t_i, t_j]$. It can increase the probability of i -type event occurring at time t (note that $g_{ij}(t) \equiv 0$ means that there is no excitation between events; the conditional intensity function degenerates to a constant; and the temporal point process becomes a Poisson process with a parameter of μ). Figure 4 is an example of a binary (two types

FIGURE 2: Examples of the sequence of keystrokes.

FIGURE 3: Triggering relationship between keystrokes of fixed text.

of events, represented by red and blue) Hawkes process. Figure 4(a) shows the calculation process $N(t)$, and Figure 4(b) shows the conditional intensity function corresponding to the blue event, conditional strength function $\lambda_2(t)$, and base intensity $\mu_2(t)$. Figure 4(c) shows the time when the two types (dimensions) of events occur.

3.3. Log-Likelihood of Samples. Suppose that the D -dimensional multivariate Hawkes process is composed of D single-

variable temporal point processes $N_i(t)$, where $i = 1, \dots, D$, and i represents dimensions. According to (2), its conditional strength function can be written as

$$\lambda_i(t|H_t) = \mu_i + \sum_{j=1}^D \sum_{\tau \in H_t^j} g_{ij}(t - \tau), \quad (3)$$

where H_t^j represents the historical event that occurred in the temporal point process j before t , and $H_t = \cup_{j=1}^D H_t^j$. The

FIGURE 4: Example of a specific bivariate Hawkes process.

exponential kernel function is a commonly used excitation kernel function:

$$g_{ij}(t) = \omega_{ij}\beta e^{-\beta t}, \quad \{t > 0\}, \quad (4)$$

where $\omega_{ij} \in \mathbb{R}_+^D$; matrix $\mathbf{W} = [\omega_{ij}] \in \mathbb{R}_+^{D \times D}$ is the adjacency matrix (or branching matrix), which describes the enhancement of the excitation intensity of the i -dimensional event by the j -dimensional event; and the attenuation coefficient $\beta \in \mathbb{R}_+$ describes the attenuation of the excitation intensity. Therefore, this article uses the adjacency matrix as the triggering matrix between keystroke events. The model parameters are $\mathbf{Z} = \{\{\mu_i\}_{i=1}^D, \{\omega_{ij}\}_{i,j=1}^D\} \in \mathbb{R}_+^{D(1+D)}$, and the attenuation coefficient $\beta > 0$ is the hyperparameter of the model.

In order to describe the sequence of keystroke events with multiple Hawkes procedures, the following defines the sequence of keystrokes. According to the definition of the multiple Hawkes process, the temporal point process i corresponds to the up or down action of the i -th character of the keystroke event. Sample keystroke behavior $\mathbf{X} = \{\{x_k^d\}_{k=1}^{n_i}\}_{d=1}^D$ represents the data observed in the sampling time interval $[0, T]$; the superscript d of $\mathbf{X}$ is the dimension of the multivariate Hawkes process; and its maximum value D is the sample corresponding number of keys. For example, if the keystroke event sequence corresponding to the “hello” text contains the last Enter key, the number of keys is 6. The subscript k of $\mathbf{X}$ is the k -th action of the temporal point process of a certain dimension. For example, the first action of the second dimension of the keystroke event sequence corresponding to the text “hello” is the down action of the button “E,” and the maximum value n_i is the number of actions in the temporal point process of this dimension (the buttons have down and up actions, so all

$n_i = 2$ in this study). A sample set consisting of multiple samples $\mathbf{S} = \{\mathbf{X}^{(m)}\}_{m=1}^M$, the superscript of $\mathbf{X}$ indicates the first m samples, and M is the total number of samples.

The log-likelihood [23] of the sample $\mathbf{X}$ is

$$\log p(\mathbf{X}|\mathbf{Z}) = \sum_{i=1}^D \left[\sum_{\tau \in H_T^i} \log \lambda_i(t|H_\tau) - \int_0^T \lambda_i(t|H_t) dt \right], \quad (5)$$

where T is the time taken for a single sample.

Then, the log-likelihood of the sample set $\mathbf{S}$ is

$$\log p(\mathbf{S}|\mathbf{Z}) = \sum_{j=1}^M \sum_{i=1}^D \left[\sum_{\tau \in H_{T_j}^i} \log \lambda_i(t|H_\tau) - \int_0^{T_j} \lambda_i(t|H_t) dt \right]. \quad (6)$$

In order to reduce the structural risk of the model, a regularization term $(1/\alpha)R(\mu, \mathbf{W})$ was added to the log-likelihood of the sample set. The maximum-likelihood estimate of the sample set $\mathbf{S}$ with regularization is (introducing another hyperparameter penalty term coefficient α)

$$\hat{\mathbf{Z}} = \arg \min_{\mu_i \geq 0, \omega_{ij} \geq 0} \left[-\log p(\mathbf{S}|\mathbf{Z}) + \frac{1}{\alpha} R(\mu, \mathbf{W}) \right]. \quad (7)$$

Here, the hyperparameter α constrains the influence of the regularization term on the likelihood estimation. In addition to the regular term constraint, in this study, we used (1) to force the constraint excitation matrix $\mathbf{W}$. Considering that the adjacency matrix is sparse, to satisfy the necessary conditions for the stationarity of the Hawkes process, the excitation function $g_{ij}(s)$ must satisfy $\int_0^\infty g_{ij}(s) ds < \infty$, and the spectral radius of the adjacency matrix spectral radius of the adjacency matrix $\mathbf{W}$ must be less than 1 to ensure the

smoothness of the model parameters. Hence, the L_2 regular term (ridge regression) was used:

$$R(\boldsymbol{\mu}, \mathbf{W}) = \sum_{i,j=1}^D \sqrt{\sum_{m=1}^M (\omega_{ij}^m)^2}. \quad (8)$$

The L_2 regular term can be described as a zero-mean Gaussian distribution on the weight ω_{ij} :

$$p_{\alpha}(\omega_{ij}) = \frac{1}{\sqrt{2\pi\alpha}} \exp\left(-\frac{\omega_{ij}^2}{2\alpha}\right). \quad (9)$$

3.4. Model Selection. The multivariate Hawkes process can be used to mine the incentive relationships that exist in the sequence of multiple types of events. For example, Eichler et al. [24] used the multivariate Hawkes process to mine the causal relationship between different types of events. The commonly used multivariate Hawkes process model training method is to add a regular term to the maximum-likelihood estimation of the sample to constrain the complexity of the model parameters to avoid overfitting. Zhou et al. [25] used regular terms with forced sparse and low-rank structures. In order to avoid excessive assumptions on the model parameters, Xu et al. [26] used the linear combination of basic functions as the intensity function and used the sparse group-lasso regular term to constrain the sparsity of the linear combination coefficients. The abovementioned frequency methods all require a large number of training samples. As the training sample size decreases, the noise of the model will become larger, so it is not suitable for keystroke behavior analysis (the sample size is not large).

Compared with frequency-based methods, Bayesian methods can also be effective when the training sample size is small. The Bayesian method uses the priors of the model parameters combined with the training data. According to the objective function (usually the maximum likelihood of the sample), the method continuously corrects and obtains the optimal parameter posterior and finally uses the parameter posterior to make decision-making inferences. As long as there is a reasonable prior, a reasonable decision can be made even if the training sample size is small. Linderman and Adams [27] used the Gibbs sampling method to approximate the likelihood of the sample. All historical events must be considered when calculating the intensity, so the convergence speed is slow. In order to improve the speed of convergence, Linderman and Adams [28] divided the time axis into many small time periods. Each function describes the excitation relationship. The function and intensity are independent, so that the calculation of intensity does not need to consider all historical events. Compared with the Gibbs sampling method [27], the convergence speed is greatly improved. However, this approach introduces model noise [29]. Salehi et al. [29] proposed a multivariate Hawkes process variational inference method. Compared with Linderman and Adams [27, 28], this method can converge quickly and simultaneously learn the model parameters and the coefficient α of the regular term, which improves the

efficiency of model learning. In this study, we used Salehi's method [29] to model the sequence of fixed text keystroke events. Like Salehi's method, we set the number of Monte Carlo samples to one; and the iterations are $I = 20000$. For each sample, the computation complexity is $O(I)$ and the runtime is about 15 minutes.

4. Experiment and Results

In order to facilitate comparison with other methods, in this study, we used the CMU dataset [1]. The dataset at <http://www.cs.cmu.edu/~keystroke/DSL-StrongPasswordData.xls> has 51 user keystrokes. Each user typed in the fixed text "tie5Roanl" 50 times once (session), and each user typed it 8 times (the interval between sessions was more than 1 day). Each user has a total of 400 (5000×8) rows of records. Each row records 1 sample X of the model. Each sample has 11 types of events (11 keys: 10 characters plus Enter key), and each type of event has 2 actions (down and up). Kevin [2] showed that, no matter which classifier was used, the accuracy of s036 and s052 was high (it is easy to distinguish from other users), and the accuracy of s002, s032, and s047 was low. This article selected the data of these 5 representative users for experiments. We determined the hyperparameter β of the Hawkes process model Code address: <https://github.com/zcmail/KD-feature-extracted-by-Hawkes-process>, trained the Hawkes process model to extract keystroke behavior characteristics, and finally compared and analyzed the raw data of CMU keystrokes and the POHMM Code address: <https://github.com/vmonaco/pohmm> model [21, 30].

4.1. Selection of Hyperparameter β of Hawkes Mode. In this study, we used grid search to select the value of β . The model was trained with samples of each session of each user, and the intensity function vector and the adjacency matrix corresponding to each session sample were obtained. The best choice of β was determined by comparing the intensity function vector and the adjacency matrix. Figure 5(a) shows the comparison of the excitation matrix of the first session sample of the s002 user under different β values; Figure 5(b) shows the comparison of the excitation matrix of the second session sample of the s032 user under different β values. Figure 5(c) shows the training result of the third session sample of the s036 user. The value of β increases (to save space, the graphs of 0.005, 0.03, and 0.05 are omitted here, which does not affect the observation of the trend), and the triggering relationship learned by the model becomes weaker (the value of the adjacency matrix is getting smaller and smaller).

According to Figure 5, the value range of β is limited to [0.001, 0.005, 0.01, 0.03, 0.05, 0.1], and the value of β is further determined according to the principle of minimizing changes within the class. The specific method is to compare the adjacency matrix corresponding to different sessions of the same user and select β corresponding to the small change of the adjacency matrix. Figure 6 shows the adjacency matrix corresponding to the samples of each session learned by the

FIGURE 5: Adjacency matrices of different users obtained based on different β values. FIGURE 6: Adjacency matrices corresponding to samples of the s002 user in different sessions obtained based on different β values.

model when the s002 user has different β values. Figure 7 shows the adjacency matrix corresponding to the samples of each session learned by the model when the s032 user has different β values. It can be seen from Figures 6 and 7 that the smaller the β value, the greater the triggering effect and the larger the change of the adjacency matrix within the class. Based on this, we further narrowed the value of β to the range of [0.01, 0.03, 0.05, 0.1].

Next, the value of β can be determined according to the principle of large differences between classes. The basic

intensities and adjacency matrices corresponding to different session samples were connected into feature vectors. Then, principal components analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) were used for reducing dimensionality and for visual analysis, respectively. Figure 8 shows the effect of different values of β .

In Figure 8, each point corresponds to the feature vector extracted from the sample data training of a certain session of a certain user, and users are distinguished by color and label (the label of user s002 is 0, the label of s032 is 1, etc.). It

FIGURE 7: Excitation matrices of the s032 user in different sessions obtained based on different β values.

can be seen from Figure 8 that when the value of β is 0.1, the discrimination effect is better. $\beta = 0.1$ was selected in this study.

Except the decay parameter β , we initialized the penalty term coefficient α in $[0.1, 1]$ and found that when α equaled one, the model sometimes could not converge. Therefore, we initialize α equal to 0.1. We use Adam optimizer and the learning rate of 0.01.

4.2. Feature Visualization. Here, the original data of CMU keystrokes (the sample is directly taken as feature data) and the feature data extracted by the Hawkes process model are visualized. Figure 9(a) shows the DD time feature of the CMU keystroke raw data of 5 users, and Figure 9(b) shows its hold time feature. The ordinate is the characteristic value; the abscissa is the sample serial number corresponding to the characteristic. The sample serial number of s002 is 0~399, the sample serial number of s023 is 400~799, and so on. It can be seen from the figure that the hold time corresponding to the sample of s036 (serial number 800~1199) is shorter than that of other users, and the DD time between *ie*, *ro*, and *nl* is longer than other users. The DD time and hold time characteristics of s002 and s032 are similar, and they are not easy to distinguish. The characteristics of s047 and s002 and s032 are similar, and the DD time between them is longer than that of other users. s052 has no features that distinguish it from other users. The results in Kevin's work [2] showed that s052 is easy to distinguish from other users.

Figure 10 shows the partial value of the intensity function vector μ extracted for 5 users using the Hawkes process model (because μ_4 , μ_5 , μ_6 , and μ_7 are 0, they are not drawn). The top

row is μ_0 , and the second row is μ_1 . The feature number is consistent with Figure 9. It can be seen from Figure 10 that, according to μ_0 , s036 is easy to distinguish from other users; according to μ_2 , s052 is easy to distinguish from other users; and, according to μ_3 , some samples of s032 can be distinguished. Figure 11 shows the partial values of the adjacency matrix extracted by 5 users using the Hawkes process model (same as Figure 10, the value 0 is not drawn). It can be seen from Figure 11 that s002, s032, and s047 are not easy to distinguish from other users; s036's $\omega_{5,4}$, $\omega_{6,5}$, and $\omega_{9,8}$ are easy to distinguish from other users; and s052's $\omega_{2,0}$, $\omega_{9,7}$, and $\omega_{10,9}$ are easy to distinguish from other users. It is difficult to distinguish s002 and s032 from Figure 9, while, in Figures 10 and 11, s002 and s032 can be distinguished according to μ_3 , $\omega_{7,5}$, and $\omega_{7,6}$. Therefore, compared with the original data of CMU keystroke events, the features extracted in this study have more specific information to distinguish users.

4.3. Comparison of Classification Results of Different Feature Data. In practical application, the positive sample data is easy to collect, while the negative sample is often unknown. We used one-class classifiers to compare the classification effects of CMU keystroke raw data, as well as the features extracted by the Hawkes process model and POHMM model (the model contains features). Kevin [2] and Ali et al. [30] used the first 200 samples of the target user to train the model. Similar to their approach, the training set used in this study comprised the first 200 samples of the target user (or the feature data corresponding to the first 200 samples) as

FIGURE 8: PCA and t-SNE visualization of features extracted based on different β values.

(a)

(b)

FIGURE 9: Original CMU keystroke data of 5 users.

FIGURE 10: Part of the values of the intensity function vector extracted by the Hawkes process model.

positive samples. Classifiers were tuned to have a 5% false alarm rate; and we compared the miss rates. Different from the works of Kevin and Ali et al., we used the remaining 200 samples of the target user taken as positive test samples, and the samples of other users (400 per user) are taken as negative test samples.

We compared the classification effects of the CMU keystroke raw data as well as the features extracted by the POHMM and the multivariate Hawkes process model. For the CMU keystroke raw data, we adopted the scaled Manhattan classifier (in [2], this classification effect is the best). The POHMM model is a generative model. It can extract the features of sample set but cannot extract the

features of single sample. The result of POHMM is the log-likelihood value of the test sample. The threshold value is selected according to the log-likelihood value of the first 200 samples of the target user (the count of log-likelihood values greater than the threshold is $5\% * 200 = 10$). The features extracted by the Hawkes process model used scaled Manhattan classifier and Euclidean classifier in [2]. These methods use the same training data and test data. Table 1 shows the classification effect. Each user has two values. The upper value is false negative, and the lower value is false positive. It can be seen that, for s002 and s032 positive samples, the classification effect of Euclidean classifier based on the features extracted by Hawkes process model is better

FIGURE 11: Excitation matrix extracted using the Hawkes process model.

than that of scaled Manhattan classifier based on CMU keystroke raw data. For the negative samples of s047, the classification effect of Euclidean classifier based on the features extracted by the Hawkes process model is better than that of scaled Manhattan classifier based on CMU

keystroke raw data and POHMM model. On the whole, the POHMM model has the best classification effect. However, no matter which classification method was used, s032, s002, and s047 have higher error rate [2]. The ROC curves of different classification results are shown in Figure 12. The

FIGURE 12: Continued.

FIGURE 12: ROC curves of different classification results. Different classification results comparison of (a) s002, (b) s032, (c) s036, (d) s052, and (e) s047.

TABLE 1: The classification effect with false alarm rate = 5%.

User	CMU raw data scaled Manhattan	POHMM	Hawkes feature scaled Manhattan	Hawkes feature Euclidean
s002	0.190	0.315	0.245	0.125
	0.291	0.140	0.354	0.436
s032	0.240	0.220	0.15	0.080
	0.314	0.428	0.574	0.549
s036	0.0	0.035	0.065	0.165
	0.0	0.0	0.011	0.0
s047	0.085	0.05	0.23	0.115
	0.475	0.121	0.331	0.125
s052	0.02	0.025	0.135	0.371
	0.0	0.002	0.491	0.344

results of the classification experiments show that it is effective to extract the features by the Hawkes process model. For users who are not easily distinguishable, this method has slightly better performance than scaled Manhattan classifier based on CMU keystroke raw data. For users such as s036 and s052 that are easy to distinguish [2], the classification effect of scaled Manhattan classifier based on CMU keystroke raw data is better than that of Euclidean classifier based on the features extracted by the Hawkes process model. Maybe noise was introduced in the training process of the Hawkes process model. In addition, the selection of

superparameter β was not accurate, which leads to the deviation of model parameters.

5. Conclusion

Taking the time interval (discrete value) between keystroke events as a feature is a common practice in modeling the sequence of keystroke events. There are a small number of studies that consider continuous time features. In this study, we used the multivariate Hawkes process based on the exponential excitation kernel to model the sequence of fixed text keystroke

events. Then, the continuous temporal characteristics of keystroke behavior were mined. The model can extract features of a single sample. By comparing the results of Kevin [2] and Ali et al. [30], we found that the features (model parameters μ and $\mathbf{W}$) learned by the model are slightly more accurate in distinguishing users that are not easily distinguishable and have richer information for distinguishing users. The exponential excitation kernel function (which is memoryless) used in this study is a strong assumption. If the keystroke behavior does not conform to the exponential decay law, there will be deviations from the actual. The next step of this study will be to explore the use of the nonparametric Hawkes process to model keystroke events (the kernel function has no specific form). In addition, since the Hawkes process can extract the features of the fixed text keystroke event sequence, it should also be able to extract the features of the free text keystroke event sequence, which can be investigated in the future work.

Data Availability

The data used to support the findings of this study are available at <http://www.cs.cmu.edu/~keystroke/DSL-StrongPasswordData.xls> (dataset address), <https://github.com/zcmail/KD-feature-extracted-by-Hawkes-process> (experiment code address 1), and <https://github.com/vmonaco/pohmm> (experiment code address 2).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. K. Kevin, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks*, pp. 125–134, Lisbon, Portugal, June 2009.
- [2] S. K. Kevin, *A scientific understanding of keystroke dynamics*, Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2012.
- [3] S. Venugopalan, J.-X. Felix, C. Benjamin et al., "Electromyograph and Keystroke Dynamics for Spoof-Resistant Biometric authentication," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 109–118, Boston, MA, USA, June 2015.
- [4] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 367–397, 2002.
- [5] J. A. Robinson, V. W. Liang, J. A. M. Chambers, and C. L. MacKenzie, "Computer user verification using login string keystroke dynamics," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 2, pp. 236–241, 1998.
- [6] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351–359, 2000.
- [7] L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabuuti, "User authentication through typing biometrics features," in *Biometric Authentication (ICBA), Volume 3071 of Lecture Notes in Computer Science (LNCS)*, pp. 694–700, Springer-Verlag, Berlin, Germany, 2004.
- [8] C. C. Epp, "Identifying emotional states through keystroke dynamics," M. S. thesis, University of Saskatchewan, Saskatoon, Canada, 201.
- [9] M. Ulinskas, M. Woźniak, and R. Damaševičius, "Analysis of keystroke dynamics for fatigue recognition," in *Proceedings of the International Conference on Computational Science and Its Applications*, Cagliari, Italy, July 2017.
- [10] W. G. de Ru and J. H. P. Eloff, "Enhanced password authentication through fuzzy logic," *IEEE Expert*, vol. 12, no. 6, pp. 38–45, 1997.
- [11] S. Mondal and P. Bours, "Person identification by keystroke dynamics using pairwise user coupling," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1319–1329, 2017.
- [12] Md Liakat Ali, J. V. Monaco, C. C. Tappert, and Q. Meikang, "Keystroke biometric systems for user authentication," *Journal of Signal Processing Systems*, vol. 86, pp. 175–190, 2017.
- [13] C.-H. Lin, J.-C. Liu, and L. Ken-Yu, "On neural networks for biometric authentication based on keystroke dynamics," *Sensors and Materials*, vol. 30, no. No.3, pp. 385–396, 2018.
- [14] K. Sung and S. Cho, "GA SVM wrapper ensemble for keystroke dynamics authentication," in *Advances in Biometrics. ICB, D. Zhang and A. K. Jain, Eds.*, pp. 654–660, Springer, Berlin, Germany, 2006.
- [15] A. Onan and S. Korukoglu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.
- [16] A. Onan, S. Korukoglu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.
- [17] A. Onan, "Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering," *IEEE Access*, vol. 7, pp. 145614–145633, 2019.
- [18] O. Alpar, "Frequency spectrograms for biometric keystroke authentication using neural network based classifier," *Knowledge-Based Systems*, vol. 116, pp. 163–171, 2017.
- [19] W. Chen and W. Chang, "Applying hidden markov models to keystroke pattern analysis for password verification," in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2004)*, pp. 467–474, Las Vegas, NV, USA, November 2004.
- [20] W. Chang, "Improving hidden markov models with a similarity histogram for typing pattern biometrics," in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2005)*, pp. 487–493, Las Vegas, NV, USA, August 2005.
- [21] J. V. Monaco Tappert and C. Charles, "The partially observable hidden markov model and its application to keystroke dynamics," *Pattern Recognition*, vol. 70, 2017.
- [22] T. Liniger, *Multivariate Hawkes Processes*, Ph.D. dissertation, ETH Zurich, Zürich, Switzerland, 2009.
- [23] E. Bacry, I. Mastromatteo, and J.-F. Muzy, "Hawkes processes in finance," *Market Microstructure and Liquidity*, vol. 1, no. 1, p. 1550005, 2015.
- [24] M. Eichler, R. Dahlhaus, and J. Dueck, "Graphical modeling for multivariate Hawkes processes with nonparametric link functions," *Journal of Time Series Analysis*, vol. 38, no. 2, pp. 255–242, 2016.
- [25] K. Zhou, H. Zha, and L. Song, "Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes

- processes,” in *Proceedings of the AISTATS*, Okinawa, Japan, June 2013.
- [26] H. Xu, M. Farajtabar, and H. Zha, “Learning granger causality for Hawkes processes,” in *Proceedings of the International Conference on Machine Learning*, Long Beach, CA, USA, June 2016.
 - [27] S. W. Linderman and R. Adams, “Discovering latent network structure in point process data,” *International Conference on Machine Learning*, vol. 32, no. 2, pp. 1413–1421, 2014.
 - [28] S. W. Linderman and R. P. Adams, “Scalable Bayesian inference for excitatory point process networks,” 2015, <http://arxiv.org/abs/1507.03228>.
 - [29] F. Salehi, W. Trouleau, M. Grossglauser, and P. Thiran, “Learning Hawkes Processes from a Handful of Events,” in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, December 2019.
 - [30] M. L. Ali, J. V. Monaco, and C. C. Tappert, “Biometric studies with hidden Markov model and its extension on short fixed-text input,” in *IEEE 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 258–264, NY, USA, October 2017.