User Identification Using Real Environmental Human Computer Interaction Behavior

Tong Wu[1*], Kangfeng Zheng[1], Chunhua Wu[1], Xiujuan Wang[2]
[1]School of Cyberspace Security, Beijing University of Posts and Telecommunications
Beijing, 100876-China
[e-mail: wutong@bupt.edu.cn]
[2]School of Computer, Beijing University of Technology
Beijing, 100124-China
[e-mail: xjwang@bjut.edu.cn]
*Corresponding author: Tong Wu

Received August 3, 2018; revised October 21, 2018; revised December 12, 2018; accepted December 28, 2018; published June 30, 2019

Abstract

In this paper, a new user identification method is presented using real environmental human-computer-interaction (HCI) behavior data to improve method usability. User behavior data in this paper are collected continuously without setting experimental scenes such as text length, action number, etc. To illustrate the characteristics of real environmental HCI data, probability density distribution and performance of keyboard and mouse data are analyzed through the random sampling method and Support Vector Machine(SVM) algorithm. Based on the analysis of HCI behavior data in a real environment, the Multiple Kernel Learning (MKL) method is first used for user HCI behavior identification due to the heterogeneity of keyboard and mouse data. All possible kernel methods are compared to determine the MKL algorithm's parameters to ensure the robustness of the algorithm. Data analysis results show that keyboard data have a narrower range of probability density distribution than mouse data. Keyboard data have better performance with a 1-min time window, while that of mouse data is achieved with a 10-min time window. Finally, experiments using the MKL algorithm with three global polynomial kernels and ten local Gaussian kernels achieve a user identification accuracy of 83.03% in a real environmental HCI dataset, which demonstrates that the proposed method achieves an encouraging performance.

Keywords: User Identification, Biometric, Multiple Kernel Learning (MKL), Keystroke Dynamic, Mouse Dynamic

http://doi.org/10.3837/tiis.2019.06.016 **ISSN : 1976-7277**

1. Introduction

A common task in data analysis is to identify users by exploiting statistics of their biometric data [1]. User identification is the process of determining who the user is [2]. In cyberspace, user identification has a wide use, such as personalized recommendations, system security, etc. The challenge of user identification is the tradeoff between maximizing user identification accuracy and minimizing its cost. The cost includes whether the equipment is highly convenient and whether the method is highly available in a real environment.

A biometric system, which is the most direct means of expressing for who the user is, relies on measurements of physiological or behavioral characteristics to establish or verify the identity of individuals [3]. In cyberspace, behavioral biometric systems rely on computer interface devices such as the keyboard and mouse that are already commonly available with most computers, and thus are of low cost in terms of having no extra equipment requirements [4].

The analysis of typing rhythms on a keyboard and mouse, which are called keystroke dynamics and mouse dynamics, have received more attention in the past few years. Keystroke dynamics refers to the process of measuring and assessing a human's typing rhythm on digital devices and it is fairly unique to each individual due to unique neurophysiologic factors [5]. As far back as the end of the 19th Century, telegraph operators at the time could often identify each other by listening to the rhythm of their Morse code keying patterns [6]. To our knowledge, Gaines *et al.* [7] are the first to investigate the possibility of using keystroke time features for authentication. Mouse dynamics involves monitoring the way a user moves the mouse in order to use that data as a means for identification and authentication [2]. As early as 2003, Gambia and Fred [8] collected mouse-movement and mouse-click data of volunteers to play a memory game on a web page for 10-15 min, and used this behavioral information to verify the identity of an individual.

In the past few years, a significant number of studies have appeared in the area of keystroke and mouse dynamics. Most published research collects the data from several volunteers and a group of features based on a controlled environment [9] to improve user identification accuracy. Numerous pattern recognition approaches, such as statistical and machine learning methods, are widely used in user identification based on keystroke and mouse dynamics [10]. However, results of these user identification methods exhibit significant differences due to various data acquisition environments and datasets, leading to difficulty in reproducing such experimental results in the actual application environment.

For actual human-computer-interaction, mouse operation and keyboard operation are integrated; that is, users complete a series of clicks and input behaviors through consequent keyboard and mouse movements. There are many studies on keystroke dynamics and mouse dynamics separately. However, a few studies have considered HCI behavior, which is the fusion of keyboard and mouse behavior data. Several existing studies about HCI behavior mainly use traditional keyboard and mouse features with shallow machine learning algorithms such as support vector machine (SVM), decision tree (DT), etc. Few of these studies consider the differences in user keyboard and mouse operation behavior and the effective methods of integrating keyboard and mouse behavior data. Another issue is that most of these studies are focused on a controlled environment and one of them achieves quite a low user identification accuracy with an uncontrolled environment dataset.

In this paper, a new user identification method is proposed in a real environment by monitoring user HCI behavior including keyboard and mouse operation in daily life. Several

analyses are carried out on keyboard and mouse data. Meanwhile, the feature fusion method is described with detailed experiments to improve user identification accuracy with valid fusion of HCI behavior data. First, 21 users' daily HCI behavior data are collected, including keyboard and mouse data lasting for several hours in their personal computer. Then, HCI features are extracted according to the results of previous research. Finally, the multiple kernel learning (MKL) method is used to identify users, accompanied by a detailed discussion of kernel parameter selection. To provide references for the proposed method, the probability density functions of all the collected data are analyzed using a random sampling method. In addition, different time intervals for dividing a sample called time windows are also compared to study the difference between keyboard and mouse data.

The rest of this paper is organized as follows. In Section 2, a brief review of related work on keystroke dynamics, mouse dynamics, fusion of keyboard and mouse data and the use of MKL algorithm in biometric identification is presented. In Section 3, we explain the framework of the user identification system in a real environment, including data collection, feature extraction, and the MKL classification algorithm. In Section 4, we focus mainly on feature distribution and performance with different time windows in a real environment. Algorithmic parameters and experimental results are presented in Section 5. Finally, in Section 6, we conclude the paper and outline planned future work.

2. Related work

2.1 Keystroke dynamics and mouse dynamics

In keystroke analysis literature, a distinction is often made between static and dynamic (or continuous) analysis [9,11]. As compared to static methods, keystroke dynamics analysis on free and long text is closer to real-world scenes. Most researchers develop their studies on free text [12-14] by setting experimental scenes such as text with almost same number of words [11,15], text concerning the same topic [16] or same equipment, etc. In addition, other researchers use totally free text collected in real environments without restrictions. As early as 2002, Dorland *et al.* [17] achieved an acceptance rate of approximately 60% with five users by monitoring their regular computing activities. Recently, Ahmed and Triode [18] presented a new approach for the free-text analysis of keystrokes that combined monograph and digraph analysis. They used a neural network to predict missing digraphs based on the relationship between monitored keystrokes. Studies on keystroke dynamics from other aspects, such as keystrokes with different input devices [19] and keystroke features [20-22], also have been carried out. Villani *et al.* [19] showed that identification accuracy decreased significantly when users used different keyboard types (desktop or laptop keyboards) or different input modes (copy or free-text input) for training and testing. Morales *et al.* [21] improved the authentication accuracy rate through feature-score normalization techniques. Wu *et al.* [22] developed a two-factor, pressure-enhanced keystroke-dynamics-based security system by converting typing motions into analog electrical signals.

After the first mouse dynamics research proposed by Gamboa and Fred [8], most common mouse dynamics techniques are centered on mouse feature extraction and classification method selection. In current studies, mouse features are divided into two categories, statistical features [8,23-25] and mouse-click features [26-28]. Statistical features calculate mouse-movement characteristics over a period of time and mouse-click features are computing features based on a single mouse click. For the studies of statistical features, Ahmed and Traore [23] introduced a definition of mouse movement action and a detailed

feature framework including movement speed, movement direction, action type, traveled distance, and elapsed time. In 2012, Feher *et al.* [25] proposed new mouse features in conjunction with features that were adopted from [8] and [23,24]. For the studies of mouse-click features, Zheng *et al.* [26] proposed an approach focused on fine-grained angle-based metrics that could distinguish a user accurately with a few mouse clicks independently. Mondal and Bours [27,28] built a continuous authentication system using a trust model denoted by the distribution of the classifier score. Results showed that all of the impostor users were identified within 344 average number of impostor actions (ANIA). In addition, some researchers take into account the mouse feature optimization problem [29,30] and achieve encouraging performance.

2.2 Fusion of keyboard and mouse data

Only a few studies exist in which researchers use a combination of keystroke and mouse dynamics for continuous authentication [31]. Ahmed and Traore [32] first proposed the use of keyboard and mouse fusion data. Early works also use graphical user interface (GUI) and stylometry features for the fusion system. Most of these studies are conducted in a controlled environment with some pre-defined tasks and use machine learning approaches for pattern classification, such as SVM, bayesian network (BN), decision tree (DT), etc. [31]. On the basis of the work of Mondal and Bours [31], previous studies of fusion identification and authentication methods are summarized in **Table 1**.

After the research of Ahmed and Traore [32], Traore *et al.* [33] introduced a risk-based authentication system for an experimental social network website that achieved an EER of 8.21% using BN models. In 2015, Wu *et al.* [34] presented an active user behavior-identification-based data-loss prevention model combining user keystroke and mouse behavior. The researches recounted above basically use traditional keystroke and mouse features. In the research of Jagadeesan and Hsiao [35], two new features were proposed, namely mouse-to-keyboard interaction ratio and interaction quotient (IQ). Apart from the fusion of keyboard and mouse data, the fusion of some other data is studied by several researchers, such as GUI and stylometry data. For example, Pusara [36] and Bailey *et al.* [2] used keyboard, mouse-movement, and GUI information in their studies, and encouraging performance was achieved by different classification algorithms. Fridman *et al.* [4] proposed a fusion architecture with keystroke dynamics, mouse movement, and stylometry. Recently, Mondal and Bours [31] analyzed a system combining continuous user authentication with identification based on user keystroke and mouse actions. They proposed that it was not possible to easily extend the results from experiments in a controlled setting. Therefore, they collected data in an uncontrolled environment with no instruction or any specific task. Finally, they obtained an identification accuracy of 62.2% for a closed-set experiment and 58.9% with an open-set experiment.

Table 1. Fusion method studies

References	Users	Methods	Performance	Environment
Bailey *et al.* [2]	31	BN, DT, SVM	Accuracy 99.39%	Controlled
Fridman *et al.* [4]	67	Naive Bayesian, SVM	FAR 0.1% and FRR 0.2%	Controlled
Mondal and Bours [31]	25	DT, Neural Network, SVM	Accuracies 58.9% and 62.2%	Uncontrolled
Ahmed and Traore [32]	22	Neural Network	FAR 1.312% and FRR 0.651%	Controlled
Traore *et al.* [33]	24	BN	EER 8.21%	Controlled
Wu *et al.* [34]	10	SVM	Accuracies 83.13%, 80.25%, and 92.64%	Controlled

Jagadeesan and Hsiao [35]	20	Neural Network, KNN	Accuracy 82.22% and 96.4%	Controlled
Pusara [36]	61	DT, SVM	FAR 23.37% and FRR 1.50%	Controlled

2.3 MKL in biometric identification

Researchers have paid significant attention to the kernel method, which has benefited from the development and application of SVM theory [37]. Since then, the kernel method has been improved, widely promoted and applied in many fields. The multiple kernel model is a kind of flexible and stronger kernel-based learning model. Recently, theory and application have proved that using multiple kernels instead of a single kernel can enhance the interpretability of decision functions and can obtain better performance than single kernel models [38,39]. The multiple kernel method has attracted the attention of researchers in many fields since it was first proposed in the field of biometrics [40,41].

Current multiple kernel methods are mainly used for image feature processing and object classification in image and video. Yang *et al.* [42] proposed a group-sensitive multiple kernel learning (GS-MKL) method for object recognition to accommodate intraclass diversity and interclass correlation. Similarly, Yeh *et al.* [43] proposed a novel MKL algorithm with a group lasso regularizer, called group lasso regularized MKL (GL-MKL), for heterogeneous feature fusion and variable selection. Althloothi *et al.* [44] used a MKL method to fuse two sets of features, namely shape representation and kinematic structure features, for human activity recognition using a sequence of RGB-D images. Later, Yan *et al.* [45] introduced a generalized adaptive lp-norm multiple kernel learning (GA-MKL) to train a robust image classifier based on multiple base kernels.

3. Proposed method

In this section, the proposed user identification method is described using HCI behavior data. Two data collection programs are developed for gathering user HCI data and extracting user HCI features, including keyboard features and mouse features. The MKL method is applied for user identification. As one of MKL methods, the Simple MKL (SimpleMKL) algorithm is adopted due to its stable performance and convenience of use.

3.1 Method description

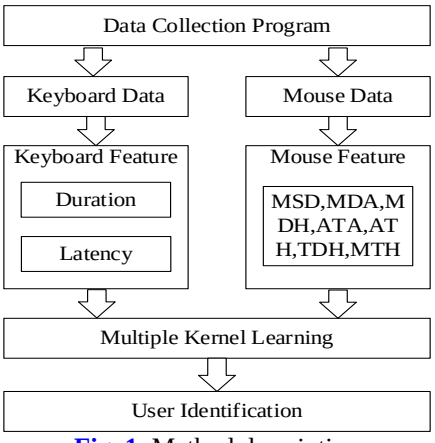

Fig. 1. Method description

Method availability in totally free environments is an important research topic for the development of user identification methods. To improve availability, a system is designed as shown in **Fig. 1**, including data collection, feature extraction, model training, and classification, to study if a user can be identified based on his daily HCI behavior, which consists of keyboard and mouse actions other than behaviors in particular scenes. In this paper, keyboard and mouse data are combined as HCI data and the MKL method is used to model and classify data for user identification.

3.2 Data collection

In order to ensure continuous data collection and tune to the necessity of method availability in real environments, a keystroke data collection program and a mouse action collection program are developed for Windows operating system. Two programs automatically run in the background when users turn on their computers, ensuring that the users' daily computer actions are collected as faithfully as possible. Twenty-one participants from our lab volunteer to deploy the data collection programs on their own machines and conduct their daily activities without any restrictions [31]. To ensure the controllability and rationality of the experimental data, these participants use their personal computers that are purchased in the same product batch of our lab. The participants follow their own routines, including taking their computers everywhere and shutting down their computers at any time.

Two separate keyboard and mouse programs collect the metadata simultaneously. The two programs gather user keyboard and mouse information by connecting to the hook chains. Four hook types are mainly used for information collection, including WH_CALLWNDPROC, WH_GETMESSAGE, WH_KEYBOARD and WH_MOUSE. The hook called WH_GETMESSAG is mainly applied for communication with the system. Keyboard metadata are obtained through WH_KEYBOARD and mouse metadata through WH_MOUSE. All the metadata are stored in several files for later processing.

Keyboard metadata include key name and timestamp of key press and key release. Mouse metadata include the number of action types, timestamp, and X and Y coordinates of the mouse pointer position. There are a total of 110 keys used by the 21 users and these keys basically cover a variety of versions of the keyboard information including letter keys, number keys, function keys, number keyboard keys, etc. An average of 199,200 keyboard records and 172,364 mouse records per user are collected to replicate the realities of users' daily HCI scenarios.

3.3 Keyboard features

Two types of features, duration time and latency time, are used most often in previous research. Duration time is the hold time of each key, which can be obtained by subtracting a key release timestamp value from its key press timestamp value. Latency time refers to the time interval between two successive keystrokes. It includes four different types of time: Up-Down (UD), Down-Down (DD), Up-Up (UU) and Down-Up (DU) time.

Using the keyboard metadata, a list of keys is obtained with its press and release timestamps. Hold time of each key is obtained by computing the mean value of all hold times in a time window. In contrast, latency time has a huge dimension and is difficult to calculate. Considering there are too many keys, for the work described in this paper a mapping matrix is created to store all the latency features. Each row $row_i (i = 1,2,...,110)$ and each column $column_j (j = 1,2,...,110)$ in the mapping matrix label all of the 110

collected keys. Each $(row_i, column_j)$ stores the four types of latency time features when the key row_i switches to the key $column_j$. The mapping matrix presented in **Table 2** facilitates the calculation of latency time features in free environments.

Table 2. Mapping matrix of latency time features

	A	B	C	...
A	(A, A)	(A, B)	(A, C)	(A, ...)
	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}
B	(B, A)	(B, B)	(B, C)	(B, ...)
	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}
C	(C, A)	(C, B)	(C, C)	(C, ...)
	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}
...	(..., A)	(..., B)	(..., C)	(..., ...)
	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}	{UD, DD, UU, DU}

A keystroke sequence S refers to a key action list closely linked in time; that is $S = \{a_1, a_2, ..., a_n\}$, where $a_i (i = 1, 2, ..., n)$ represents a key action. A user's keystroke behavior K consists of multiple keystroke sequences; that is, $K = \{s_1, s_2, ..., s_m\}$, where $s_i (i = 1, 2, ..., m)$ represents a keystroke sequence. The problem is distinguishing a new keystroke sequence according to the length of latency time as a result of continuous automatic data acquisition in a real environment. Users' rests and computer shutdown behaviors cause a lot of data blanks, which increase the difficulty of data pre-processing and render judgement difficulty whether a user engages in a temporary departure or a long time leave from his work according to the length of blank time. Therefore, according to the dataset, a threshold is set to deal with this problem. A new keystroke sequence starts when latency time is greater than 1 min and a flag marking a user's shutdown behavior is used to start a new keystroke sequence.

Every sample derives 110 duration time features and 48,400 latency time features with 110 keys. If there is more than one action of the same key in a time window, the average duration and latency time values are calculated as the feature value. Because most key transfers will never occur, features that never have a value assigned for any user are removed as they do not contribute to the classification algorithm [2]. Relief is a feature selection algorithm used in binary classification proposed by Kira and Rendell in 1992 [46]. Kononenko *et al.* [47] proposed a new method to extend binary classification to multi-classification, in which the weight of each feature was obtained through the Relief function in MatLab (MathWorks, USA). Finally, over 4000 keyboard features are used in total, of which all the weights are greater than 0.

3.4 Mouse features

The mouse features are derived from Ahmed and Traore [23]. Four types of actions including mouse-move (MM), drag-and-drop (DD), point-and-click (PC) and silence are calculated to classify user mouse actions and provide uniform feature standards. DD means the action starts with mouse button down, movement, and then mouse button up. Silence means there is no movement over a period of time. Eight directions [23] numbered from 1 to 8 are proposed

as shown in **Fig. 2** based on mouse pointer coordinates on the computer screen. The angle is the offset of the line beginning with the mouse cursor starting point and end with the terminal point.

The mouse dynamics features consist of seven types organized by five categories: movement speed, movement direction, action type, traveled distance, and elapsed time. The seven types include movement speed compared to traveled distance (MSD), average movement speed per movement direction (MDA), movement direction histogram (MDH), average movement speed per types of actions (ATA), action type histogram (ATH), traveled distance histogram (TDH), and movement elapsed time histogram (MTH). In the work described in this paper, a total of 60 mouse features are used in the experiments.

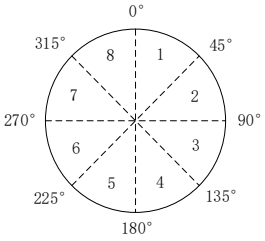

Fig. 2. Eight directions of mouse movement [23]

3.5 Simple Multiple Kernel Learning (SimpleMKL)

Keyboard and mouse actions exhibit different performance; that is, keystrokes are composed of single keys and mouse actions are composed of mouse movement and click. Different sources and composition of data increase the difficulty of data combination and identification. It is thus important to find the appropriate combination method.

The MKL method has been extensively researched and applied in the fields of classification, multi-class object detection and recognition, pattern regression, and feature extraction. The combination of kernel functions is an inevitable choice to meet some practical requirements, such as heterogeneous information or unnormalized data, large scale problems, non-flat distribution of samples, etc [37]. Noble [48] calls this method of combining kernels intermediate combination and contrasts it with early combination (where features from different sources are concatenated and fed to a single learner) and late combination (where different features are fed to different classifiers, the decisions of which are then combined by a fixed or trained combiner). In this work, fusion data of keyboard-time and mouse-action features are particularly heterogeneous and unnormalized since two kinds of features reflect different pieces of useful information with over 4,000 features involved. Multiple kernels instead of a single kernel can enhance the interpretability of fusion data and help the classifier achieve a high accuracy rate.

SimpleMKL as provided by Rakotomamonjy is employed for the work described in this paper. It addresses the MKL problem through a weighted 2-norm regularization formulation with an additional constraint on the weights that encourages sparse kernel combinations and solves a standard SVM optimization problem, in which the kernel is defined as a linear combination of multiple kernels [49].

In such cases, a convenient approach is to consider that the kernel $K(x,x')$ is actually a convex combination of basis kernels:

$$K(x,x') = \sum_{m=1}^{M} d_m K_m(x,x'), \quad with\ d_m \geq 0,\ \sum_{m=1}^{M} d_m = 1 \tag{1}$$

where M is the total number of kernels. Each basis kernel K_m can either use the full set of variables describing x or subsets of variables stemming from different data sources [50]. Alternatively, the kernels K_m can simply be classical kernels (such as Gaussian kernels) with different parameters. Within this framework, the problem of data representation through the kernel is then transferred to the choice of weights d_m [49].

The SimpleMKL algorithm is proposed to address the MKL-based SVM problem by solving the convex problem defined as follows.

$$\begin{aligned} &\min_{d}\max_{\alpha} \quad J(d,\alpha)=-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_m d_m K_m(x_i,x_j)+\sum_i \alpha_i \\ &s.t. \quad \sum_i \alpha_i y_i = 0, \quad C \ge \alpha_i \ge 0 \quad \forall i, \\ &\qquad \sum_m d_m = 1, \quad d_m \ge 0 \quad \forall m \end{aligned} \tag{2}$$

In this paper, two basis kernel functions are used, a polynomial function and a Gaussian function, the latter of which is another form of radial basis function (RBF). Each, with different parameters, is linearly combined to classify users. The RBF, Gaussian function, and polynomial function are defined as follows.

$$K_{RBF}(x,z)=\exp(-g\cdot\|x-z\|^2) \tag{3}$$

$$K_{Gaussian}(x,z)=\exp(-\frac{\|x-z\|^2}{2\sigma^2}) \tag{4}$$

$$K_{Poly}(x,z)=\langle x,z\rangle^d \tag{5}$$

Here, σ is the bandwidth of the Gaussian function and d is the degree of the polynomial function. In LibSVM, a RBF is used and the parameter g is equal to $1/2\sigma^2$ in a Gaussian function. The use of three kernel methods including parameter selection and kernel combination is elaborated in Section 5.

4. Feature analysis

For the lack of HCI feature analysis in free environment, keyboard and mouse feature distribution and time window performance are compared in this paper. The HCI behavior data of 21 users are collected in daily life to extract keyboard and mouse behavior features according to previous research. The distribution of keyboard and mouse data are analyzed using the random sampling method. At the same time, to achieve better identification results, classification performances of keyboard and mouse data with different time windows are compared using the SVM algorithm.

4.1 Feature distribution analysis

In recent years, more researchers have been attracted to the analysis of keyboard and mouse dynamics; however, few papers are focused on the analysis of keyboard and mouse features. In this paper, the data distribution of daily keyboard and mouse features is studied to illustrate the differences between keyboard and mouse features. The probability distribution and probability density function of the collected keyboard and mouse data are shown in **Fig. 3** and **Fig. 4** as a result of using the random sampling analysis method. **Figs. 3** and **4** show that the distribution of mouse data is flatter than that of keyboard data. The parameter σ representing the distribution width is 103.9 of keyboard data probability distribution, while that of mouse data probability distribution is 817.3. The two types of features exhibit a huge

diversity in figures and parameter values. Distribution of keyboard data is concentrated mainly within 500 and distribution of mouse data falls into two parts, one-part around 0 and the other more dispersed. The appearance of data distribution can be explained by the different methods of feature extraction. The mouse features consist of count data and frequency distribution data, which causes great differences in numerical distribution from one decimal place to a thousand. The keyboard features are all of the time statistical data and users' key press time is relatively concentrated together in daily HCI behavior. Analyses show that keyboard and mouse data belong to heterogeneous data; therefore, new methods are needed for user identification. Ordinary classification algorithms that only combine keyboard and mouse features do not necessarily achieve the best results.

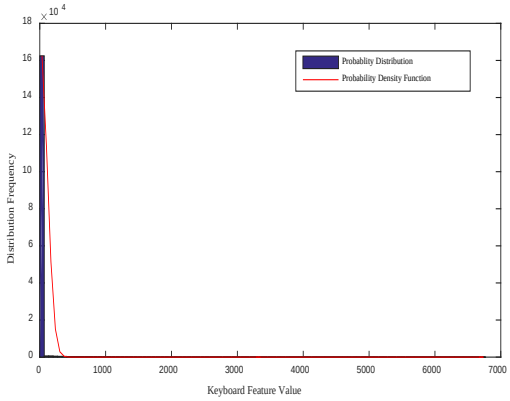

Fig. 3. Keyboard data distribution

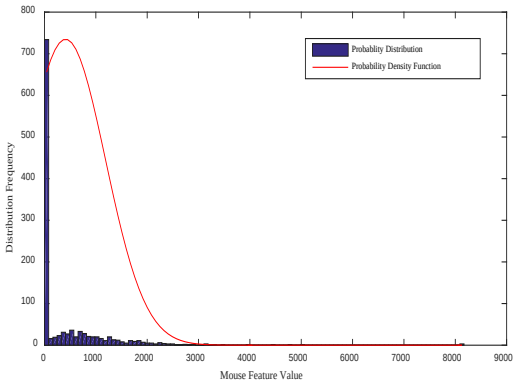

Fig. 4. Mouse data distribution

4.2 Feature performance analysis with different time window

To illustrate the difference between keyboard and mouse behavior, in this paper we concentrate on the performance of different time windows, which refers to the time interval for dividing a sample. Bailey *et al.* [2] adopted a 10-min time window in their study, but they did not explain this choice. Owing to the different structures of keyboard and mouse features, time window selection will affect the accuracy of user identification and the fusion method of keyboard and mouse data. In this paper, time windows of 1 to 10 min are compared using keyboard operation data, mouse operation data, and their fusion data.

According to grid-SVM, sample number and the best accuracy of keyboard and mouse data with different time windows from 1 to 10 min are shown in **Fig. 5**. With increasing time window, sample number of keyboard and mouse data becomes smaller. Moreover, the classification accuracy of keyboard data decreases, while that of mouse data increases. The entirely opposite performance illustrates that the change of accuracy has little to do with the number of samples, and the more important factor is the different feature extraction methods of keyboard and mouse data. As time window increases, more computer actions are available and thus lead to different consequences of keyboard and mouse data. Keyboard features are all statistical time of different keyboard actions, while mouse features are the distribution of different mouse actions. More keyboard actions make the average keystroke time of each user more equal, therefore making the difference between users more indistinct. In contrast, more mouse actions make the mouse action distribution more differentiated. **Fig. 6** shows sample number and classification accuracy tendencies of the fusion data with different time windows. There are relatively fewer samples number with a 1-min time window due to the narrow time interval. With growth of the time window, the classification accuracy rate increases at the beginning, reaches a peak in a time window of 4-min, and then shows a

downward trend. This trend is caused by differences in keyboard and mouse features and will have an effect on the performance of user identification method.

According to the performance of different types of data, different time windows are selected for different data. Three time windows, 1, 10, and 4 min, are assigned to keyboard, mouse and fusion data, respectively.

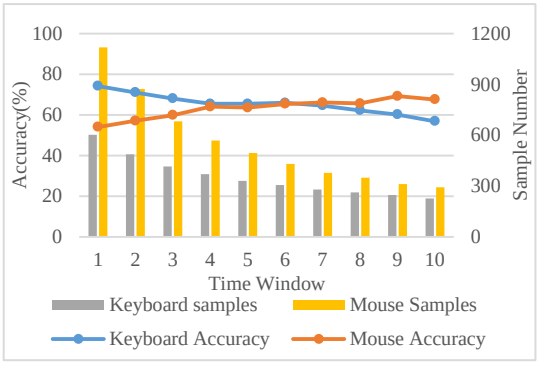

Fig. 5. Keyboard and mouse data performance with different time windows

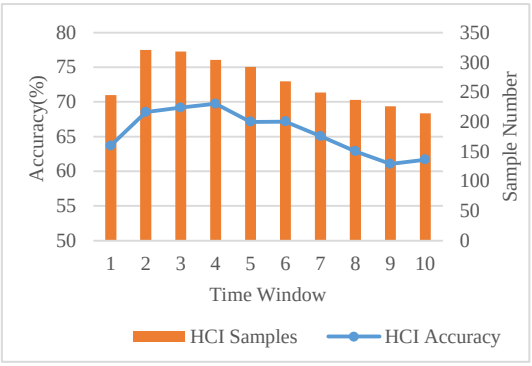

Fig. 6. Fusion data performance with different time windows

5. Experimental results

Three types of experiments are carried out. First, appropriate kernels for MKL are chosen by comparing the performance of single kernels and different kernel combinations. Then, 3 h of training data and 1 h of testing data are used to verify the method availability in a real environment. Finally, eight common user identification algorithms are compared using the measurement of identification accuracy. All experiments are done using LibSVM and the SimpleMKL toolbox in MatLab 2015b.

5.1 Kernel selection

To choose appropriate candidate kernels for the MKL algorithm, we compare the performance of polynomial kernel functions of different degrees, and give the best c and g values of the RBF kernel function through a grid optimization algorithm using LibSVM. Then, different combinations of kernel functions are compared to determine the actuating boundary of the polynomial and RBF kernels. All results are obtained by a 10-fold cross-validation method on 3 h of users' data.

Different degrees from 1 to 5 are performed with a polynomial kernel. **Table 3** shows the classification results of different data with various degrees. For mouse data with a 10-min time window, the highest accuracy value appears in a degree of 4, and there is no significant change in the overall trend. For the keyboard data with a 1-min time window and fusion data with a 4-min time window, the highest values both occur in a degree of 1. With increasing degrees, the accuracy rates of both types of data show very rapid attenuation. The polynomial kernel function of 1 degree is equal to the linear kernel function, indicating that keyboard and fusion data exhibit the best performances with a linear kernel function.

Table 3. Classification results of polynomial kernel

Polynomial degree	1	2	3	4	5
Keyboard(1-min)	**73.926**	48.4652	17.0392	12.4492	11.4695
Mouse(10-min)	59.4023	60.7241	62.7126	**62.8276**	60.3563
Fusion(4-min)	**62.5269**	45.0323	27.9892	20.3978	14.7634

Table 4 shows the best accuracy rates with a RBF kernel and the optimal *c*, *g*, σ values, the last of which is the bandwidth of the Gaussian kernel function. Compared with the polynomial kernel function, the results of SVM with a RBF kernel have high identification accuracy among keyboard, mouse and fusion data. This illustrates that the RBF kernel is able to interpret user HCI data better with appropriate parameters.

Table 4. Best accuracy with RBF kernel

	Kernel	*c*	*g*	σ	Accuracy (%)
Keyboard(1-min)	RBF	27.8576	0.0039	11.3	**74.1379**
Mouse(10-min)	RBF	22.3786	0.4835	1	**67.5768**
Fusion(4-min)	RBF	9.1896	0.0039	11.3	**69.7368**

On the basis of the results presented in **Tables 3** and **4**, the best performance of a single kernel and the corresponding parameters are found. The kernels that have the best performance will be chosen as the basis of a multiple-kernel approach. Polynomial kernels of degree $d \in \{1,2,3\}$ and Gaussian kernels with $\sigma \in \{1,2,5,7,10,11.3,12,15,17,20\}$ are candidate kernels. Thus, 13 alternatives are obtained for parameterizing the two defined kernel functions. As different kernels have different performance with various feature sets, polynomial kernels and Gaussian kernels with global feature sets and local feature sets are integrated in this work. Eight combinations based on two kinds of kernels are derived with a basis kernel number ranging from 13 to 23.

Table 5 shows different accuracy results for different multiple kernel methods with fusion data. All of the measurements are acquired by taking the average of results by five 10-fold cross validation experiments. Different combinations of polynomial kernels and Gaussian kernels are compared in **Table 5** with variables for random and all. Random means that the kernel acts on local feature data, while all means the kernel acts on the global feature data. Different variables of the same combination of kernels are also compared in **Table 5**. Accuracy, standard deviation(SD), coefficient of variation(CV) and time consumed are used as comparative measurements. It is shown that global polynomial kernels together with local Gaussian kernels have the best performance. This can be interpreted to mean that a Gaussian kernel function has the ability to extract local feature information, which means that it is sensitive to local information, while the polynomial kernel function is sensitive to global data.

Table 5. Results achieved with different multiple kernel methods

Kernel	Variable	Accuracy (%)	SD	CV	Time(s)
Polynomial, Gaussian	Random, Random	69.19	9.71	14.03	126.99
Polynomial, Gaussian	All, All	73.74	7.14	9.69	235.43
Polynomial, Gaussian	**All, Random**	**74.39**	**6.89**	**9.26**	**118.95**
Polynomial, Gaussian	Random, All	68.29	10.10	14.78	256.39
Polynomial, Gaussian, Polynomial	Random, Random, All	73.61	9.41	12.78	147.20

Polynomial, Gaussian, Gaussian	Random, Random, All	67.60	10.18	15.07	389.94
Polynomial, Gaussian, Gaussian	All, Random, All	73.49	8.49	11.55	299.51
Polynomial, Polynomial, Gaussian	All, Random, All	73.69	7.87	10.68	232.16

5.2 Evaluation results

After determining all the parameters using 3 h of user behavior training data, 1 h of behavior data are used as testing data to verify the usability of the proposed method. The 21 users work on HCI tasks lasting for 1 h, and 165 testing samples with a 4-min time window are collected. To compare the final identification results, 391 keyboard testing samples with a 1-min time window and 94 mouse testing samples with a 10-min time window are also collected in 1 h. Over 4,000 keyboard features and 60 mouse features as detailed in Sections 3.3 and 3.4 are applied in this experiment.

Table 6 lists the final identification results using 3 h of training samples and 1 h of testing samples. An accuracy of 83.03% is achieved by HCI data combining keyboard and mouse features using the MKL algorithm, while accuracies of 79.54% and 74.47% are achieved by keyboard and mouse data, respectively, using the SVM algorithm. The final identification results show that HCI data using the MKL method perform better than single keyboard and mouse data. On the basis of the kernel methods compared in Section 5.1, the improvement of accuracy is mainly derived from the use of multiple kernel learning methods, which proves the effectiveness and availability of the proposed method.

Table 6. Identification results

Data	Number of training samples	Number of testing samples	Number of correct samples	Accuracy (%)
HCI	304	165	137	83.03
Mouse	293	94	70	74.47
Keyboard	603	391	311	79.54

The classification results distributions for each user with the MKL method using HCI data are shown in **Fig. 7**. The results of keyboard and mouse data using the SVM single kernel method are shown in **Fig. 8** and **Fig. 9**, respectively. In **Figs. 7-9**, the Y axis represents each user and the X axis represents where the samples belong to according to the classifier. The numbers in the matrix represent the percentage of each user's HCI samples judged as others. Therefore, it represents the recall rate for each user along the diagonals. Recall rate indicates the number of each user's correctly classified samples relative to the total sample number of each user as shown in formula (6). Here, i is the number of users, TP_i the correctly classified sample number of each user, and P_i the total sample number of each user. The recall rate shows the performance of all the samples of each user. The higher the recall rate, the darker the color is.

$$Recall_i = TP_i / P_i \tag{6}$$

As **Fig. 8** shows, keyboard performance differs greatly among each user. For example, users 1 and 2 have low recall rates and there is no high rate in the first and second rows. This illustrates that the user 1 and 2 samples have no obvious distinguishing attributes compared with other users. For user 5, 56% of the samples are classified as user 16, illustrating that user 5 samples are similar to user 16 samples. The mouse data performance as shown in **Fig.**

9 displays similarity with the keyboard data, of which some users have poor performance. **Fig. 7** illustrates that the proposed method improves and balances the identification accuracy for each user through a unified classification model. Intuitively, the performance distribution of HCI data is closer to the performance of keyboard data due to the large number of keyboard features. Meanwhile, the performance distribution of HCI data also takes advantage of mouse features and improves the performance of individual keyboard features, demonstrating the advantage of the MKL algorithm and the proposed method.

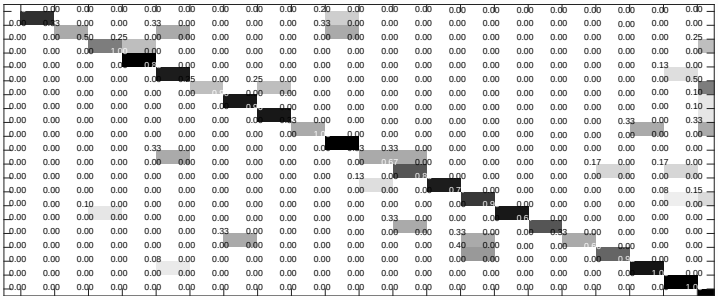

Fig. 7. Classification result distribution using HCI data

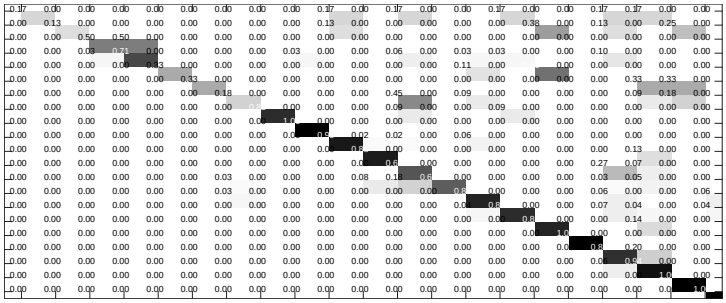

Fig. 8. Classification result distribution using keyboard data

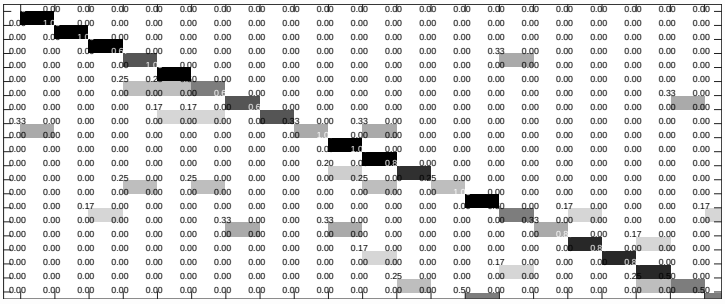

Fig. 9. Classification result distribution using mouse data

5.3 Method comparison

According to Section 2.2, seven common machine learning algorithms used in previous works are compared with the MKL method proposed in this paper, including Random Forest(RF), SVM, BN, k-nearest neighbor (KNN), NaiveBayes(NB), DT and Neural Network(NN). It is worth noting that it is difficult to repeat all the experiments in previous

research owing to the differences in experimental techniques. These differences include environmental set (limited or free), device flexibility, the amount of data in training and testing, etc. Because of these discrepancies, in the work described in this paper we only use the seven algorithms in training and testing samples shown in Section 5.2 for a concise reference. **Fig. 10** clearly illustrates the classification result of all eight algorithms. The MKL method proposed in this paper achieves the best performance, while the others show variable results, indicating that some machine learning algorithms are not suitable for heterogeneous information or large scale problems in free environments. The proposed method is thus adapted to the needs of user identification in free environments.

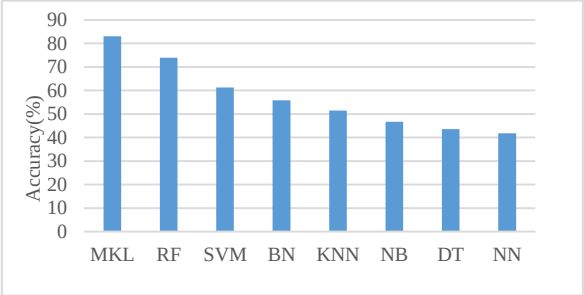

Fig. 10. Methods comparison

6. Conclusions and planned future work

In this paper, a user identification method based on real environmental HCI behavior is presented by analyzing the differences between keyboard and mouse data in real environments through comparing different kernel methods. The HCI data of 21 users are collected using two programs running on their personal computers for 4 h. HCI features are extracted according to previous research, including keyboard and mouse features. Probability distributions of HCI data are worked out through the random sampling method and the performances of different time windows are obtained by the SVM algorithm. Several experiments are conducted to determine algorithm parameters and verify method availability. Finally, user identification results obtained using the MKL method are presented and compared with the results of other machine learning algorithms.

Experimental results show that the data distributions of keyboard and mouse behavior are very different and the performances of different time windows of dividing samples are also different; more concretely, the best time windows of fusion, keyboard and mouse data are 4, 10, and 1 min, respectively. By comparing different kernel methods, the data show that the combination of 3 global polynomial kernels and 10 local Gaussian kernels achieves the best performance. For separate keyboard and mouse data, the SVM algorithm with a RBF kernel performs best, while for the fusion data, the simpleMKL algorithm has the best result, which is higher than the identification results of separate data. By training with 3 h of data and testing with 1 h of data, an accuracy of 83.03% is finally obtained using the method proposed in this paper, which is higher than that of the keyboard data (79.54%) and that of the mouse data (74.47%). Comparison with other algorithms using the same samples also shows the good performance of the proposed method.

There are in total just a few studies of user identification in real environments, and the identification accuracy achieved in this study is promising. However, it needs to be improved in real environment applications. While factors for user identification and feature extraction

of heterogeneous data will be considered in our planned future studies. The extension of our approach to other behavior data, such as the web behavior data, will be considered in our later work.

Acknowledgements

This work is supported by the National Science and Technology Major Project under Grant no.2017YFB0802800 and the National Natural Science Foundation of China under Grant no.61602052.

References

[1] Naini, F. M., Unnikrishnan, J., Thiran, P., and Vetterli, M, "Where You Are Is Who You Are: User Identification by Matching Statistics," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 358–372, February, 2016. Article (CrossRef Link).

[2] Bailey, Kyle O., James S. Okolica, and Gilbert L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77-89, June, 2014. Article (CrossRef Link).

[3] Michael Fairhurst, Meryem Erbilek, and Márjory Da Costa-Abreu, "Selective review and analysis of aging effects in biometric system implementation," *IEEE transactions on human-machine systems*, vol. 45, no. 3, pp. 294-303, June, 2015. Article (CrossRef Link).

[4] Fridman, L., Stolerman, A., Acharya, S., Brennan, P., Juola, P., Greenstadt, R., and Kam, M., "Multi-modal decision fusion for continuous authentication," *Computers & Electrical Engineering*, vol. 41, pp. 142-156, January, 2015. Article (CrossRef Link).

[5] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, pp. 1-24, 2013. Article (CrossRef Link).

[6] John Leggett, Glen Williams, Mark Usnick, and Mike Longnecker, "Dynamic identity verification via keystroke characteristics," *International Journal of Man-Machine Studies*, vol. 35, no. 6, pp. 859-870, December, 1991. Article (CrossRef Link).

[7] Gaines, R. Stockton, William Lisowski, S. James Press, and Norman Shapiro, "Authentication by Keystroke Timing: Some Preliminary Results," *RAND Corporation*, 1980. Article (CrossRef Link).

[8] Gamboa Hugo and Ana LN Fred, "An Identity Authentication System Based On Human Computer Interaction Behaviour," in *Proc. of International Workshop on Pattern Recognition in Information Systems*, vol. 2003, pp. 46-55, April, 2003.

[9] Salil Partha Banerjee and Damon Woodard, "Biometric Authentication and Identification Using Keystroke Dynamics: A Survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116-139, July, 2012. Article (CrossRef Link).

[10] Md Liakat Ali, John V. Monaco, Charles C. Tappert, and Meikang Qiu, "Keystroke Biometric Systems for User Authentication," *Journal of Signal Processing Systems*, vol. 86, no. 2-3, pp. 175-190, March, 2016. Article (CrossRef Link).

[11] Daniele Gunetti and Claudia Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312-347, August, 2005. Article (CrossRef Link).

[12] Tomer Shimshon, Robert Moskovitch, Lior Rokach and Yuval Elovici, "Continuous Verification Using Keystroke Dynamics," in *Proc. of International Conference on Computational Intelligence and Security*, pp. 411-415, December, 2010. Article (CrossRef Link).

[13] Paulo Henrique Pisani and Ana Carolina Lorena, "Emphasizing typing signature in keystroke dynamics using immune algorithms," *Applied Soft Computing*, vol. 34, pp. 178-193, September, 2015. Article (CrossRef Link).

[14] Fabian Monrosea and Aviel D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351-359, February, 2000. Article (CrossRef Link).
[15] Fabian Monrose and Aviel Rubin, "Authentication via keystroke dynamics," in *Proc. of the 4th ACM conference on Computer and communications security*, pp. 48-56, April 01-04, 1997. Article (CrossRef Link).
[16] Arik Messerman, Tarik Mustafic, Seyit Ahmet Camtepe and Sahin Albayrak, "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," in *Proc. of 2011 International Joint Conference on Biometrics (IJCB)*, October 11-13, 2011. Article (CrossRef Link).
[17] P. S. Dowland, S. M. Furnell and M. Papadaki, "Keystroke analysis as a method of advanced user authentication and response," *Security in the Information Society*, pp. 215-226, 2002. Article (CrossRef Link).
[18] Ahmed A. Ahmed and Issa Traore. "Biometric recognition based on free-text keystroke dynamics," *IEEE transactions on cybernetics*, vol. 44, no. 4, pp. 458-472, April, 2014. Article (CrossRef Link).
[19] M. Villani, C. Tappert, Giang Ngo, J. Simone, H.St. Fort and Sung-Hyuk Cha, "Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions," in *Proc. of 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, June 17-22, 2006. Article (CrossRef Link).
[20] Pin Shen Teh, Shigang Yue and Andrew B.J. Teoh, "Improving keystroke dynamics authentication system via multiple feature fusion scheme," in *Proc. of 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, pp. 277-282, June 26-28, 2012. Article (CrossRef Link).
[21] Aythami Morales, Elena Luna-Garcia, Julian Fierrez and Javier Ortega-Garcia, "Score normalization for keystroke dynamics biometrics," in *Proc. of 2015 International Carnahan Conference on Security Technology (ICCST)*, pp. 223-228, September 21-24, 2015. Article (CrossRef Link).
[22] Changsheng Wu, Wenbo Ding, Ruiyuan Liu, et al., "Keystroke dynamics enabled authentication and identification using triboelectric nanogenerator array," *Materials Today*, vol. 21, no. 3, pp. 216-222, April, 2018. Article (CrossRef Link).
[23] Ahmed Awad E. Ahmed and Issa Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on dependable and secure computing*, vol. 4, no. 3, pp. 165-179, July, 2007. Article (CrossRef Link).
[24] Youssef Nakkabi, Issa Traore and Ahmed Awad E. Ahmed, "Improving mouse dynamics biometric performance using variance reduction via extractors with separate features," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 6, pp. 1345-1353, November, 2010. Article (CrossRef Link).
[25] Clint Feher, Yuval Elovici, Robert Moskovitch, Lior Rokach and Alon Schclar, "User identity verification via mouse dynamics," *Information Sciences*, vol. 201, pp. 19-36, October, 2012. Article (CrossRef Link).
[26] Nan Zheng, Aaron Paloski and Haining Wang, "An efficient user verification system via mouse movements," in *Proc. of the 18th ACM conference on Computer and communications security*, pp. 139-150, October 17-21, 2011. Article (CrossRef Link).
[27] Soumik Mondal and Patrick Bours, "Continuous authentication using mouse dynamics," in *Proc. of 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, September 5-6, 2013.
[28] Soumik Mondal and Patrick Bours, "A computational approach to the continuous authentication biometric system," *Information Sciences*, vol. 304, pp. 28-53, May, 2015. Article (CrossRef Link).
[29] C. Shen, Z. Cai, X. Guan, H. Sha and J. Du, "Feature Analysis of Mouse Dynamics in Identity Authentication and Monitoring," in *Proc. of 2009 IEEE International Conference on Communications*, pp. 1-5, June 14-18, 2009. Article (CrossRef Link).

[30] Chao Shen, Zhongmin Cai, Xiaohong Guan and Jinpei Cai, "A hypo-optimum feature selection strategy for mouse dynamics in continuous identity authentication and monitoring," in *Proc. of 2010 IEEE International Conference on Information Theory and Information Security*, pp. 349-353, December 17-19, 2010. Article (CrossRef Link).
[31] Soumik Mondal and Patrick Bours, "Combining keystroke and mouse dynamics for continuous user authentication and identification," in *Proc. of 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, February 29-March 2, 2016. Article (CrossRef Link).
[32] A.A.E. Ahmed and I. Traore, "Anomaly intrusion detection based on biometrics," in *Proc. of the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop*, June 15-17, 2005. Article (CrossRef Link).
[33] Issa Traore, Isaac Woungang, Mohammad S. Obaidat, Youssef Nakkabi and Iris Lai, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," in *Proc. of 2012 Fourth International Conference on Digital Home*, November 23-25, 2012. Article (CrossRef Link).
[34] Jain Shing Wu, Chih Ta Lin, Yuh Jye Lee and Song Kong Chong, "Keystroke and mouse movement profiling for data loss prevention," *Journal of Information Science & Engineering*, vol. 31, no.1, pp. 23-42, January 2015.
[35] Harini Jagadeesan and Michael S. Hsiao, "A novel approach to design of user re-authentication systems," in *Proc. of 2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, September 28-30, 2009. Article (CrossRef Link).
[36] Maja Pusara, "An Examination of User Behavior for Re-authentication," *PhD thesis, Center for Education and Research in Information Assurance and Security, Purdue University*, August, 2007.
[37] Hong-Qiao WANG, Fu-Chun SUN, Yan-Ning CAI, Ning CHEN and Lin-Ge DING, "On multiple kernel learning methods," *Acta Automatica Sinica*, vol. 36, no. 8, pp. 1037-1050, September, 2010. Article (CrossRef Link).
[38] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui and Michael I. Jordan, "Learning the Kernel Matrix with Semi-Definite Programming," *Journal of Machine Learning Research*, vol. 5, pp. 27-72, January, 2004.
[39] Wan-Jui Lee, Sergey Verzakov and Robert P. W. Duin, "Kernel Combination Versus Classifier Combination," in *Proc. of International Workshop on Multiple Classifier Systems*, pp. 22-31, May 23-25, 2007. Article (CrossRef Link).
[40] Paul Pavlidis, Jason Weston, Jinsong Cai and William Noble Grundy, "Gene functional classification from heterogeneous data," in *Proc. of the fifth annual international conference on Computational biology - RECOMB '01*, pp. 249-255, April 22-25, 2001. Article (CrossRef Link).
[41] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan and W. S. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," *Biocomputing*, pp. 300-311, December, 2003. Article (CrossRef Link).
[42] Jingjing Yang, Yonghong Tian, Ling-Yu Duan, Tiejun Huang and Wen Gao, "Group-sensitive multiple kernel learning for object recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2838-2852, May, 2012. Article (CrossRef Link).
[43] Yi-Ren Yeh, Ting-Chu Lin, Yung-Yu Chung and Yu-Chiang Frank Wang, "A Novel Multiple Kernel Learning Framework for Heterogeneous Feature Fusion and Variable Selection," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 563-574, June, 2012. Article (CrossRef Link).
[44] Salah Althloothi, Mohammad H. Mahoor, Xiao Zhang and Richard M. Voyles, "Human activity recognition using multi-features and multiple kernel learning," *Pattern Recognition*, vol. 47, no. 5, pp. 1800-1812, May, 2014. Article (CrossRef Link).
[45] Shengye Yan, Xinxing Xu, Dong Xu, Stephen Lin and Xuelong Li, "Image Classification With Densely Sampled Image Windows and Generalized Adaptive Multiple Kernel Learning," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 381-390, March, 2015. Article (CrossRef Link).

[46] Kenji Kira and Larry A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. of the tenth national conference on Artificial intelligence*, pp. 129-134, July 12-16, 1992.
[47] Igor Kononenko, Edvard Šimec and Marko Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, no. 1, pp. 39-55, January 1997. Article (CrossRef Link).
[48] William Stafford Noble, "Support vector machine applications in computational biology," *Kernel methods in computational biology*, pp. 71-92, 2004.
[49] Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu and Yves Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2491-2521, November, 2008.
[50] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626-2635, May, 2004. Article (CrossRef Link).

Tong Wu is a Ph.D. student in the Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include network security, machine learning. Email: wutong@bupt.edu.cn

Kangfeng Zheng received his PhD degree of Information and Signal Processing in July 2006 at Beijing University of Posts and Telecommunications. He is currently an associate professor at School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests include networking and system security, network information processing and network coding. Email: zkf_bupt@163.com

Chunhua Wu is currently an instructor lecturer in the Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include network and information security, machine learning. Email: wuchunhua@bupt.edu.cn

Xiujuan Wang received her PhD degree of Information and Signal Processing in July 2006 at Beijing University of Posts and Telecommunications. She is currently an instructor lecturer at Faculty of Information Technology, Beijing University of Technology. Her research interests include informaion and signal processing, network security and network coding. Email:xjwang@bjut.edu.cn