Hindawi
Security and Communication Networks
Volume 2021, Article ID 6669429, 19 pages
https://doi.org/10.1155/2021/6669429

Research Article

Design and Implementation of Continuous Authentication Mechanism Based on Multimodal Fusion Mechanism

Jianfeng Guan,[1,2] **Xuetao Li**,[1,2] **and Ying Zhang**[1]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Jianfeng Guan; jfguan@bupt.edu.cn

Received 16 October 2020; Revised 8 July 2021; Accepted 3 September 2021; Published 15 October 2021

Academic Editor: Karl Andersson

Most of the current authentication mechanisms adopt the “one-time authentication,” which authenticate users for initial access. Once users have been authenticated, they can access network services without further verifications. In this case, after an illegal user completes authentication through identity forgery or a malicious user completes authentication by hijacking a legitimate user, his or her behaviour will become uncontrollable and may result in unknown risks to the network. These kinds of insider attacks have been increasingly threatening lots of organizations, and have boosted the emergence of zero trust architecture. In this paper, we propose a Multimodal Fusion-based Continuous Authentication (MFCA) scheme, which collects multidimensional behaviour characteristics during the online process, verifies their identities continuously, and locks out the users once abnormal behaviours are detected to protect data privacy and prevent the risk of potential attack. More specifically, MFCA integrates the behaviours of keystroke, mouse movement, and application usage and presents a multimodal fusion mechanism and trust model to effectively figure out user behaviours. To evaluate the performance of the MFCA, we designed and implemented the MFCA system and the experimental results show that the MFCA can detect illegal users in quick time with high accuracy.

1. Introduction

With the vigorous development of 5G, IoT (Internet of Things), and AI (Artificial Intelligence), the Internet has penetrated into various traditional industries, which brings in greater data privacy disclosure and more serious information security risks due to the endogenous security issues of the Internet. As the first line of network defense, authentication mechanism becomes a crucial way to ensure information security [1]. The current authentication schemes can be classified into four kinds: (1) authentications based on passwords or PINs (Personal Identification Numbers), (2) portable smart card or token-based authentications, (3) biometric-based authentications, such as face, fingerprint, and iris recognition, which are also called hard biometric-based authentications, (4) behaviour-based authentications such as gait and keystroke, which are called soft biometrics [2, 3]. More specifically, the hard and soft biometric-based authentications overcome the problems that password authentication is long and hard to remember and the problems related to smart card, which is easy to be stolen. On the contrary, biometric-based authentications do not require the authentication entity to be carried along at all times, which is inconvenient and also easy to be lost. Biometric authentications are based on human physiological behaviour characteristics, and have the advantages of natural nonreplication, which greatly improves user experience and reduces the risk of privacy disclosure [4]. However, physiological feature recognition generally relies on specific feature recognition devices such as face recognition device and fingerprint collector, which depend on expensive equipment and even poses the risk of forgery when they lack effective supervision. Besides, due to the limitations of user devices, computation and storage of authentication procedure are generally offloaded to edge or remote cloud, which may increase the attack surface and security risks.

On the other hand, the first three kinds of authentications belong to one-time authentication from the perspective of the identification mode, which only verifies the users' identity when the devices are unlocked for the first time. Once the users have passed the authentication, which is just like getting the device's pass card, they can use the system resources continuously without receiving the verification again [5]. For example, when a legitimate user temporarily leaves for tea or has a short conversation with others, the device that is not immediately locked may be at risk of being used to steal information by an adversary. These kinds of attacks can be classified as internal attacks, which are difficult to defend. The recent report from Cybersecurity Insiders [6] shows that 68% of organizations feel moderately to extremely vulnerable to insider attacks. To prevent insider attacks, the concept of "Zero Trust" has been proposed, which follows the principle of "Never Trust, Always Verify" [7]. The authentication, especially the continuous authentication, plays an import role in zero trust architecture.

This paper designs and implements a continuous authentication system that continuously monitors the user's operations after the device is unlocked. Once the system finds that user identity is abnormal, the device will be automatically locked to prevent the risk of "one-time authentication" and guarantee the user information security. The main contributions of this paper are as follows:

(1) We propose a multimodal fusion mechanism for multidimensional behaviour characteristics. Considering that the single mode recognition is not enough to effectively depict the user's behaviours, we design a multimodal fusion mechanism based on multidimensional features, and construct a trust model for continuous identity authentication, to improve the authentication accuracy and recognition rate.

(2) We select three users' behaviours to realize the multimodal identification, which include keystroke behaviours, mouse movement, and application usage characteristics. The keystroke and mouse movement are time-sensitive and convenient, and the application usage based on logs is more stable and efficient. More important, all of them do not rely on additional hardware devices, and therefore have the advantages of being low cost and user friendly.

(3) We design a Multimodal Fusion-based Continuous Authentication (MFCA) system based on multimodal fusion mechanism. The MFCA system mainly consists of three parts: First, the multidimensional behaviour models (keystroke model, mouse model, and application model) are obtained by training on multidimensional behaviours data; second, the multidimensional model classification results are fused; third, the multidimensional behaviour models are evaluated based on the trust model algorithm, and the real identity of users is evaluated based on the trust evaluation.

The structure of this paper is organized as follows. In Section 2, the related work has been discussed. Section 3 describes Multimodal Fusion mechanism and recognition models used in the MFCA. The design and procedure of the MFCA system have been discussed in Section 4. The performance of the proposed MFCA system is analysed comprehensively in Section 5. Finally, Section 6 concludes the work of this paper.

2. Related Work

The related researches in terms of authentications based on keystroke, mouse movement, or swipe and application usage are shown in Figure 1.

Keyboard and mouse, as the most commonly used input devices, have their own advantages to depict users' behaviour characteristics. Keyboard dominates text input while mouse is more commonly used in GUI. The identification based on single input device will affect the immediacy and accuracy of user identity verification. Moreover, as the inherent hardware equipment, the keyboard and mouse are transparent to users during identity verification, which can avoid targeted destruction or forgery by identity counterfeiters in advance. When users interact with the operating system using the keyboard and mouse, they will trigger the iterative update procedure of the application state. Different users' preferences for application reflect users' using habits, which are irreplaceable and can be used as an important way for identity verification.

Keystroke dynamics refers to the physiological neural control mechanism of humans, which reflects the unique characteristics by analysing users' habits, patterns, or rhythms through the keystroke such as different time intervals between keystrokes and keystroke strength. As early as the 1980s, research studies [8–10] had proved the utility of keystrokes in terms of identity verification.

Mouse dynamics refers to the track and the click of the mouse during user interaction with the system, and the most commonly used features include mouse keystroke speed, habits, frequency, and direction of the mouse moving distance. Everitt and Mcowan [11] found and proved the feasibility of knowing a user's identity by analysing the user's mouse operation habit and behaviour characteristics. With the development of computer GUI, the mouse has superseded the keyboard and become the dominant I/O device [12].

User application usage refers to the characteristic of terminal device in terms of resources scheduling, consuming, and even interacting with other equipment, which can be obtained through the system interface or process information since they are independent of special hardware. More recently, the behaviour-based continuous authentication technologies have been widely active in many fields. For example, user identity recognition can be based on smartphone applications [13], the information from sensors such as gyroscope and magnetometer [14], the users' arm movement records on smart watch [15], and gait recognition

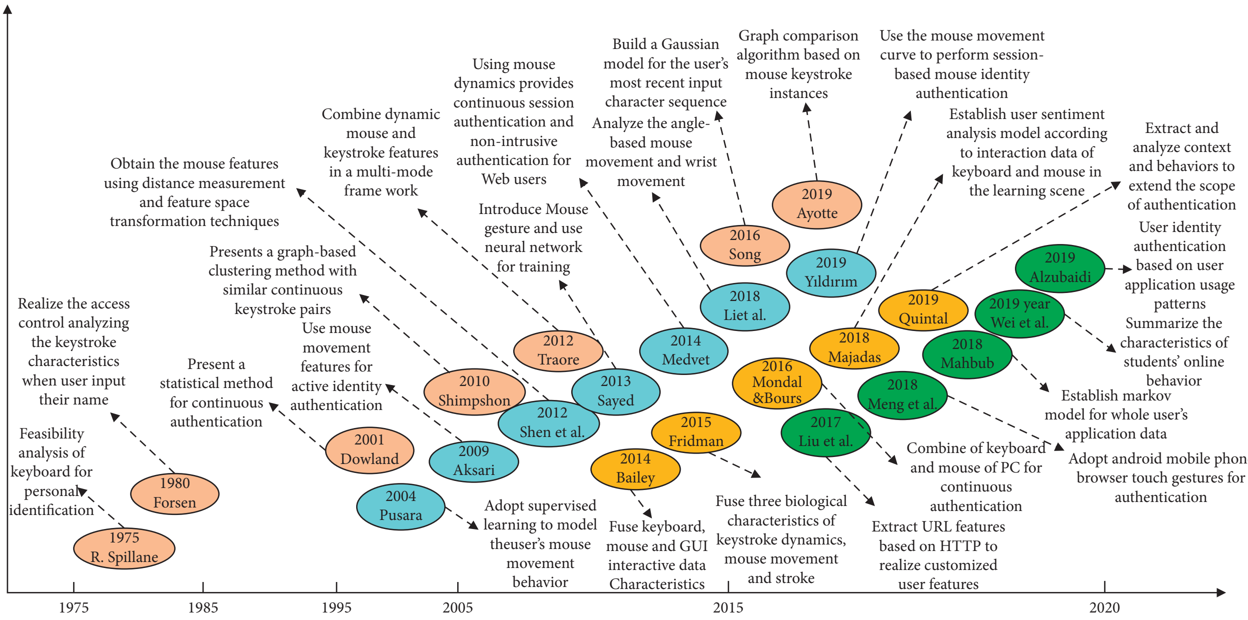
Build a Gaussian model for the user's most recent input character sequence
Graph comparison algorithm based on mouse keystroke instances
Use the mouse movement curve to perform session-based mouse identity authentication
Using mouse dynamics provides continuous session authentication and non-intrusive authentication for Web users
Combine dynamic mouse and keystroke features in a multi-mode frame work
Analyze the angle-based mouse movement and wrist movement
Establish user sentiment analysis model according to interaction data of keyboard and mouse in the learning scene
Extract and analyze context and behaviors to extend the scope of authentication
Obtain the mouse features using distance measurement and feature space transformation techniques
Introduce Mouse gesture and use neural network for training
User identity authentication based on user application usage patterns
Presents a graph-based clustering method with similar continuous keystroke pairs
2019 Ayotte
2016 Song
2019 Yildırım
2018 Liet al.
2019 Alzubaidi
Realize the access control analyzing the keystroke characteristics when user input their name
Use mouse movement features for active identity authentication
2019 Quintal
2019 year Wei et al.
Summarize the characteristics of students' online behavior
2012 Traore
2014 Medvet
2018 Majadas
2018 Mahbub
Present a statistical method for continuous authentication
2016 Mondal &Bours
2010 Shimpshon
2013 Sayed
2018 Meng et al.
Establish markov model for whole user's application data
2012 Shen et al.
2015 Fridman
2017 Liu et al.
Feasibility analysis of keyboard for personal identification
2001 Dowland
2009 Aksari
Combine of keyboard and mouse of PC for continuous authentication
Adopt android mobile phone browser touch gestures for authentication
1980 Forsen
2014 Bailey
2004 Pusara
Adopt supervised learning to model theuser's mouse movement behavior
Fuse keyboard, mouse and GUI interactive data Characteristics
Fuse three biological characteristics of keystroke dynamics, mouse movement and stroke
Extract URL features based on HTTP to realize customized user features
1975 R. Spillane
1975
1985
1995
2005
2015
2020

Figure 1: The related research of keystroke, mouse/swipe, and App usage. The different shapes represent the different authentication methods.

based on wristband [16]. While in traditional PC, the most commonly used biometric authentication is still based on keystroke and mouse [17].

2.1. Keystroke-Based Authentication Schemes. As early as 1975, Spillane [8] discussed the feasibility of keystroke in user identification, and this suggestion was also verified by Forsen and Gaines in 1977 [9] and 1980 [10], respectively. Forsen et al. [9] realized access control by analysing the keystroke characteristics of users when they input names that are similar to human signatures, while Gaines et al. [10] recorded the keystroke interval when the typist input the specified text, and analysed the time probability of consecutive typing characters, and verified the uniqueness of individual keystroke characteristics. This is the starting point of keystroke recognition, and belongs to static authentication technology based on fixed text. Fixed text refers to the predefined text or phrases to register a user, and it requires the user to type exactly the same text to perform identity verification with the objective of reducing uncertainty by controlling variables and observing the performance of a single keystroke feature in identification. This kind of experiment is generally applicable to scientific researches [18, 19], which has great limitations and is currently only applicable to the identity verification of fixed user names and passwords. Therefore, it is also called static authentication.

On the contrary, in continuous authentication scenarios, users are free to type texts that are not limited by the predefined contents. Free-text-based keystroke recognition is more difficult than fixed one in terms of data preprocessing, feature selection, and keystroke authentication [20]. Dowland et al. [21] proposed a statistical method for continuous certification in 2001, whose accuracy was less than 60%. However, after nearly ten years of development, Shimpshon et al. [22] proposed a clustering method based on graph in 2010 which added a similar continuous keystroke to form a fixed length of the session, and the experimental results in 21 real users and 165 counterfeiters showed that it has a False Accept Rate (FAR) of 3.47% just by using 250 keystrokes. More specifically, Rybnik et al. [23] explored keystrokes in different lengths of nonfixed text in 2013, providing a reliable basis for the authentication of free text. After that, Song et al. [24] constructed a Gaussian model for the user's recent input characters sequence based on the Gaussian probability density function in 2016, which shortened the authentication cycle and reached FAR of 5.3% under 30 characters. Huang et al. [25] updated the keystroke samples using the sliding window method and achieved FAR of 1% and False Reject Rate (FRR) of 11.5% in a 1-minute sliding window in 2017. Furthermore, they evaluated the ability of the proposed algorithm to resist short quick insider attacks and detected insider attacks that lasted 2.5 minutes or longer with a probability of 98.4%. More recently, Ayotte et al. [26] proposed an instance-based graph comparison algorithm, which achieved an EER (Equal Error Rate) of 7.9%, 5.7%, 3.4%, and 2.7%, respectively, under the samples of 50, 100, 200, and 500 keystrokes, realizing faster and more accurate free-text keystroke identification.

2.2. Mouse-Movements-Based Authentication Schemes. Mouse dynamics has also received attention in terms of authentication. In 2003, Everitt and Mcowan [11] proved the feasibility of mouse behaviour characteristics in user identity authentication for the first time, which set a solid foundation for the subsequent extensive researches in the academic community. In 2004, Pusara and Brodley [27] used supervised learning to model the mouse movement behaviour of 11 users and obtained FAR of 0.43% and FRR of 1.75%. However, due to the small number of user samples and single mouse features, the authors also pointed out that the analysis of that study was not enough to achieve independent user identity authentication. In 2009, Aksari and Artuner [28] used mouse movement characteristics for active identity authentication, and obtained FAR of 5.9% and FRR of 5.9% by analysing the mouse trajectory when the user clicked 10 squares in a row. In 2013, Sayed et al. [29] introduced mouse gestures into the user registration system and performed training through the neural network to realize identity verification when the user logins, and finally reached FAR of 5.26% and FRR of 4.59% within 26.9 s in a dataset of 39 users. The above researches are also called static mouse authentication, which mainly explores the diversity of mouse features and the wide application field of mouse recognition through specifying user behaviour or limiting mouse operation scope and trajectory.

On this basis, mouse movement-based continuous authentication schemes have attracted more and more attention. In 2012, Chao Shen et al. [30] evaluated 5550 mouse operation samples of 37 users, and obtained mouse features based on distance measurement and feature space transformation technology, which reached FRR of 8.74% and FAR of 7.69% within 11.8 s. Besides, they established the first public mouse-behaviour dataset, and their research results revealed the potential of mouse dynamics in user authentication. In the same year [31], the pattern-based growth method was used to mine frequent mouse behaviour fragments, and obtained more stable mouse features and reached FAR of 0.37% and FRR of 1.12%. In 2014, Medvet et al. [32] used mouse dynamics to provide continuous session authentication and nonintrusive authentication for web users, and achieved the accuracy of 97% for 24 users. Their work extended the potential scope of mouse dynamics as a continuous authentication tool to web applications hosted in the cloud rather than just in local devices. In 2018, Li et al. [33] used the random forest and sequential sampling analysis to analyse the angle-based mouse movement and wrist movement, and reached FAR of 1.46% on the dataset of 26 users, and the verification time could be determined within 9–12 mouse clicks. Their approach is more effective in timely authentication compared with methods based on the mouse geometry and locomotion features. In 2019, Yildirim and Anarim [34] verified on Balabit dataset [35] that mouse movement curves alone and session-based mouse identity could be used, achieving Area Under Curve (AUC) of 93% and EER of 13%.

2.3. Application-Usage-Based Authentication Schemes. Different from keyboard and mouse-based authentication schemes, application-usage-based authentication is not a biometric technology but a behavioural analysis based on user activity records with the objectives to mine user activity records, extract user multi-attribute behaviour characteristics, and then build user behaviour model to represent user identity and complete continuous authentication. Lots of current research efforts have proved the uniqueness of user behaviour, and thus derived a lot of user behaviour analytical methods based on big data.

In 2017, Liu [36] extracted URL characteristics and identified user's consumption level by taking users' surfing time as the sample. In 2018, Mahbub et al. [13] built a Markov model based on the complete application data of users, carried out continuous authentication by evaluating the changes of hidden Markov model (HMMs), and finally realized the capture of abnormal users within 2.5 minutes on the experimental dataset. They solved the active authentication problem by using application usage formulaically and systematically. Furthermore, they suggested that unknown application and unforeseen events had more important impacts on the authentication performance than the most common ones. In 2018, Meng et al. [37] conducted user authentication based on the touch gestures of Android mobile phone browser, achieving an average EER of 2.4% among 48 participants, and their system can reduce the touch behavioural deviation than others. In 2019, Wei [38] analysed the DNS logs of campus network by categorizing the domain names to obtain users' online behaviour habits and access preferences, and summarized the characteristics of students' online behaviours. In terms of user analysis based on application records, it can be divided into single application based, top *n* applications based, and all applications based behaviour identification by considering potential unique user behaviour pattern when using applications.

Besides, Alzubaidi et al. [39] presented an active authentication based on the smartphone usage data under different machine learning models, and achieved a lower EER of 8.2% for authenticating users within short periods of time with a small number of features on the MIT dataset [40]. Their scheme was effective in reducing the classification error rate compared with other authentication methods. For mobile devices, they are easy to deploy authentication based on the App using record, phone usage record, and even web browser history. However, due to the different operating systems, it is difficult to achieve the intersystem authentication.

2.4. Multimodal-Fusion-Based Authentication Schemes. With the development of various authentication technologies, some researchers try to combine different authentication technologies to increase the accuracy and timeliness. Although no single authentication technology is perfect, it is very difficult to fool multiple authentication methods at the same time. Therefore, multimodal fusion authentication can overcome the problems of partial feature loss. The recent work from Modak and Jha [41] summarized the multi-biometric fusion strategy and its different applications in terms of multi-modal, multi-algorithm, multi-sample, multi-sensor, and multi-instance, and proved the performance upgrade of combining two or more individual biometric traits.

As early as in 2012, Traore et al. [42] proposed an online authentication system under web environment, which combined dynamic mouse and keystroke features in a multimodal framework to conduct real-time monitoring of 24 user operations, and the final system EER was 8.21%. However, their results had a low Average Number of Genuine Actions (ANGA) value which made the system not practical for real users. In 2014, Bailey et al. [43] proposed a user authentication system based on multimodal behavioural biometrics by fusing user data from keyboard, mouse, and GUI interactions, and adopted ensemble classification method to get FAR of 2.1% and FRR of 2.24% over the dataset with 31 users, which supports the idea of multimodal fusion to gain better consequence. In 2015, Fridman et al. [44] presented a multimodal fusion for continuous authentication by collecting the behavioural biometrics of keystroke dynamics, mouse movement, and a high-level modality of stylometry, and developed a sensor for each modality and organized these sensors as parallel binary decision fusion architecture. Their experimental results based on database of 67 users who work individually for a week show that FRR and FAR are less than 1% within 30 s. In 2016, Mondal and Bours [45] proposed a continuous identity authentication for PC users by combining keystroke and mouse dynamics, and the recognition rate reached 62.6% and 58.9% in closed and open environments, respectively. The average operation times were 471 and 333, respectively. Besides, they first introduced the issue of Continuous Identification (CI) and discussed the concept of Continuous Authentication and Identification that provided the combination of security and forensics. In the same year, Beserra et al. [46] applied the dynamic identity recognition application by combining keyboard and mouse for the first time in online games, and carried out real-time identification of player operations to realize anti-cheating function. In 2018, Sergio et al. [47] established a user emotion model based on the interaction data of the keyboard and mouse in the learning scenario, so as to predict the affective state of the learner. In 2019, Quintal et al. [48] analysed the mobile user continuous authentications in IoT, and classified these authentication factors into event capture types such as password, fingerprint, applications start and end, network connection and disconnection, continuous sequence of events, such as gestures, and derived behavioural features, such as application choice, and demonstrated that all factors are correlated with the actual user identity. Currently, lots of multimodal continuous authentications are proposed in smartphone, IoT [49–51].

The key points of multimodal fusion continuous authentication are the association, unified representation, and coordination of multimodal information, and the main issues are: (1) multimodal characteristic expression, that is, how to design single-modal characteristics under the framework of multimodal architecture; (2) how to unify the model of multimodal characteristics. In our preliminary work, we have studied continuous authentication based on users' keystroke and mouse behaviour [19], and developed a prototype system. Among them, a static authentication algorithm based on convolutional neural network is proposed for user keystroke behaviour. The average accuracy on CMU dataset is 96.8%, the average FAR is 0.04%, and the average FRR is 6.5%. At the same time, a continuous authentication algorithm based on the weighted reward and punishment mechanism was proposed. When the effective double key pairs of each user are 100, the EER is 8.5% and the AUC is 93.94%.

For this purpose, this paper designs a multimodal fusion continuous authentication mechanism based on users' multidimensional behaviour characteristics in terms of keystroke, mouse, and application usage to effectively prevent the illegal user identity phishing, avoid data privacy, improve authentication efficiency, and ensure the safety of user information.

3. Multimodal-Fusion-Based Continuous Authentication

The MFCA system consists of multimodal fusion mechanism, trust model, and multidimensional behaviour recognition models. In this section, we will introduce the multimodal fusion mechanism and three recognition models that are used in the MFCA.

3.1. Multimodal Fusion Mechanism.

The multidimensional behaviours of network users mainly include keystrokes, mouse, screen swipe, and application usage. This paper designs the multimodal fusion mechanism to collect user behaviour data, combines these multidimensional features effectively, fuses the multiple classifier to avoid the limitation of the single classification and improve the classification accuracy and generalized capability, and finally realizes the continuous authentication.

3.1.1. Multi Classifier Fusion Mechanism.

Considering the diversity, complexity, and fusibility of the features, this paper adopts the Multi-Classifier Fusion (MCF) mechanism to improve the accuracy and generalization capability of the final classification results by integrating the output classification results of base classifiers. At the same time, MCF can simplify classify design, balance classification time and performance, and improve time and space efficiency. The typical structure of the MCF includes cascade combination, parallel combination, and mixed combination. Parallel combination does not have the error accumulation problem of cascade combination. Furthermore, the system can achieve the best performance of real-time classification by designing an appropriate decision process. So, this paper adopts parallel combination to perform parallel processing on the user's multidimensional behaviours including keystroke dynamics, mouse movement, and application usage data.

The results of the MCF algorithm depend on the output type of the base classifier. When the base classifier output is an interval value or probability value, we can adopt the mean value method (simple average or weighted average), maximum-minimum value method, product method, etc. When the output is a predefined class label, we adopt the voting method such as weighted voting, supermajority voting, or relative majority voting.

3.1.2. Trust Model Design. The trust model is the base of the MFCA and its time-variant characteristic is the key to realize the continuous authentication. The basic idea of the trust model is that the degree of credibility of the current operating user depends on the deviation between the user's behaviour characteristics and the expected characteristics of the model over a period of time. The system predefines trust score and trust threshold at the outset, and then increases or decreases the trust score along with their operations. When the user's behaviour characteristics conform to the model, the trust score will increase (no more than maximum score). Otherwise, the trust score will decrease. Once the score falls below the predefined trust threshold, an exception alarm will be triggered.

Figure 2 shows the schematic diagram of the fluctuating trust score. Over a period of continuous operations, the legitimate user behavioural characteristic is the most trusted attribute even though it may not be stable most of the time. The corresponding trust score will be slightly up and down in a certain period of time, but it is always higher than the trust threshold. The legitimate users will almost imperceptibly perceive the authentication system in order to ensure transparency. On the contrary, the abnormal operations of the illegal user will inevitably lead to the continuous decline of the trust score, which will eventually make the trust score lower than the trust threshold and trigger the abnormal alarm. Therefore, without relaxing the timely detection of illegal users, the design of the trust model increases the tolerance of legitimate users' misoperations to improve the accuracy and user-friendliness of the authentication system.

Table 1 shows the related parameters used in the trust model. Each user has an initial trust score of T_0. The model verifies the current user's identity status in real time according to the user's behaviour characteristic F_i. When the user's identity is judged to be legitimate, the trust model gives rewards to increase the trust score until the highest threshold $T_{\max}$. Otherwise, it will reduce the trust score until the minimum threshold $T_{\min}$. When the score is lower than the trust threshold T_{alert}, the system alarm will be triggered to lock the device. The increase or decrease of the trust score is limited by the maximum reward score R and maximum punishment score P, and the increase or decrease range depends on the reward and punishment weight W_i of the current characteristic.

According to the above definitions, we can deduce equations (1) and (2), from which the trust score T_i is obtained after the initial authentication.

$$\Delta_T(F_i) = \begin{cases} F_i * W_i * R, & F_i = 1, \\ (F_i - 1) * W_i * R, & F_i = 0, \end{cases} \tag{1}$$

$$T_i = \min\{\max\{T_{i-1} + \Delta_T(F_i), T_{\min}\}, T_{\max}\}. \tag{2}$$

The basic component of the trust model is the user's single behaviour characteristic, and the reward and punishment range of the trust score depend on the reward and punishment weight of the given characteristic. The specific weight is introduced for that the system involves three types of classification models.

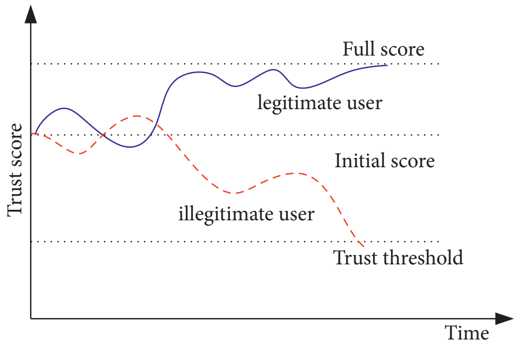

Figure 2: Schematic diagram of trust score fluctuation curve.

Table 1: The parameters of trust model.

Parameters	Meaning	Value

F_i	The classification result of i^{th} feature	{1, 0}
		Legal = 1,
		illegal = 0

T_i	The trust score of i^{th} authentication	$[T_{\min}, T_{\max}]$
$T_{\min}$	The minimum trust score	$T_{\min}$
$T_{\max}$	The maximum trust score	$T_{\max}$
T_{alert}	The alert threshold	$[T_{\min}, T_{\max}]$
W_i	The punishment weight of feature i	[0–1.00]
R	The maximum punishment score	$[0, T_{\max}]$
P	The minimum punishment score	$[0, T_{\max}]$

Different behavioural operations will generate different characteristics, but user behaviours have a certain pattern. Therefore, the characteristic with high frequency will be considered more stable and identifiable. In contrast, the characteristics with low frequency generally have lower credibility in the trust model. Take mouse keystroke events as an example; mouse keystroke events are divided into left click, left double click, right click, and right double click. When the occurrence probability of left-click events is much greater than that of right-click events, the stability of left-click behaviour is stronger, and the reward and punishment weight obtained are also higher. For example, left-click occurs 67 times, double-click occurs 20 times, right-click occurs 10 times, and double right-click occurs 3 times. When the user is judged as a legitimate user in the left-click feature, the trust score should be rewarded with 67 R/100. Otherwise, when the user is judged as an illegal user, the trust score will be punished with 67 P/100. Therefore, the trust score value is mainly affected by two major factors in the weight design: the weight ratio of characteristic model in model fusion and the frequency ratio of the feature in the feature set.

3.2. Keystroke Recognition Model

3.2.1. Keystroke Dataset Capture Module. In this section, we give the keystroke capture procedure of Windows as an example. The user interacts with the computer through the keyboard to finish the input, so the keystroke data capture range is global events. Therefore, we adopt the keyboard hook to collect the keystroke data and encapsulate hook into the Dynamic Link Library (DLL) to ensure the automatic loading and real-time collection of keystroke data. The implementation of keyboard hook is divided into three parts: the installation of keyboard hook, the monitoring and processing of keyboard message, and the uninstalling of keyboard hook. Figure 3 shows the procedure of keystroke data capture.

First, the keystroke capture module adds the keyboard hook to the list and binds the keystroke event to the keyboard hook via the *SetWindowsHookEx* () function that mainly consists of four parameters. The first parameter *idHook* represents the installed hook type which has two kinds. This module selects a global keyboard hook called *WH_KEYBOARD_LL*, which contains lots of keyboard information such as virtual keyboard key value *vKCode*, keystroke state *WM_KEYUP* and *WM_KEYDOWN*, and so on. The second parameter LPFN points to the hook subroutine for further processing of the hooked message, which is also called the call back function. In this module, we rename this function as *KeyboardProc*. The third parameter *hMod* is the current instance handle which is also known as DLL module handle. The fourth parameter *dwThreadId* is the thread identifier associated with the keyboard subroutine.

Second, the *KeyboardProc* function is used to monitor keyboard messages, and Table 2 shows the related field information to be collected. When a user clicks a key, the keyboard hook captures the event and begins to record the keystroke value, keystroke timestamp, keystroke event type, and so on.

As for the conversion of key values and codes, the commonly used ASCII codes distinguish the key values of upper and lower case letters "*a-z*" from "A-Z", with 65–90 representing uppercase letters and 97–122 representing lowercase letters. *VkCode*, on the other hand, is treated as the same keyboard key without distinction, and only records the A-Z key value with 65–90. Therefore, when collecting records, the system needs to further determine whether the Shift key is being pressed through *GetAsyncKeyState* (), and obtain the state of CapsLock key through *GetKeyState* (). When either of them is pressed, the letter key is defined as uppercase state, and vice versa.

After that, the specific type of keystroke event is obtained through *wParam*. The system aims to intercept *WM_KEYUP* (key press down) and *WM_KEYDOWN* (key release), and records the keystroke timestamp through *GetLocalTime* function. In this case, the time can be accurate to milliseconds. Finally, the *CallNextHookEx* () function is used to complete the delivery to the next hook in the list, and the keyboard hook is destroyed once the data collection is completed.

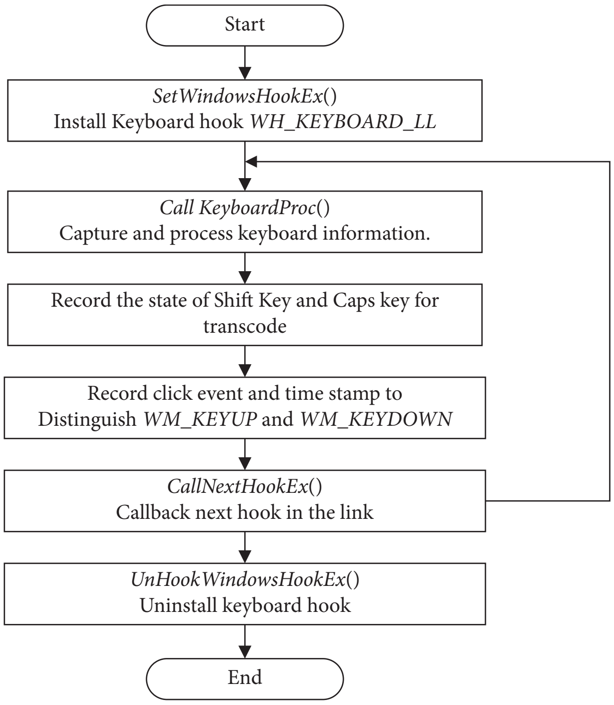
Start
SetWindowsHookEx()
Install Keyboard hook WH_KEYBOARD_LL
Call KeyboardProc()
Capture and process keyboard information.
Record the state of Shift Key and Caps key for transcode
Record click event and time stamp to Distinguish WM_KEYUP and WM_KEYDOWN
CallNextHookEx()
Callback next hook in the link
UnHookWindowsHookEx()
Uninstall keyboard hook
End

Figure 3: The procedure of keystroke capture.

Table 2: Keystroke data information.

Field name	Type	Description
keyCode	Int	Keystroke key code
keyValue	Char	Keystroke key values
keyEvent	Int	Event type
keyStamp	Long	Keystroke timestamp
isShiftOn	Bool	Shift key on/off
isCapsOn	Bool	CapsLock key on/off

3.2.2. Keystroke Data Preprocessing and Feature Selection. The original keystroke record obtained through data acquisition is a combination of key code, key value, event type, and timestamp, such as 87, W, WM_KEYDOWN, and 59108278, respectively. Due to different event types, the keystroke behaviour of the same key value distinguishes two records of press and release. Therefore, it should be merged and converted into key code, key value, press timestamp, and release timestamp at first, and then deletes the record with the missing value. Finally, the raw data are transformed into feature data.

In free-text environment, user keystroke is affected by the language, profession, and even emotional stress. Therefore, the behaviour habit is random and diverse. On one hand, the keyboard layout is complicated, which includes typical QWERTY keyboard with 87, 104, and 109 keys. On the other hand, the use of the function keys is adventitious, and its characteristics need long-term observation. Therefore, timeliness is insufficient when it is used in continuous authentication. Our system extracts the characteristics of user's inputted characters. 26 character keys will randomly form different character sequences which are affected by language grammar and common words, and the typical character combinations have a wide range of universality. When different users hit the same character, they will show different time characteristics and keystroke frequency. Besides, the length of character sequence determines the order and the magnitude of the combined sequence and the space-time loss complexity of feature processing. Therefore, we select the user double key combination comprehensively, that is, the character sequence of length 2 is used as the feature sample of the keystroke recognition model. To select the most popular double key pairs, we conduct statistical experiments on the statistical frequency of double key combinations, and record the frequency of the top 20 double key combinations among hundreds of thousands of valid character keystrokes [19], as shown in Table 3. Finally, the top 7 double keys are selected as the double key feature samples.

The seven double bond characteristics (“AN,” “NG,” “IN,” “SH,” “EN,” “IA,” and “CH”) are extracted uniformly for three types of time characteristics: Hold time, Down-Down time, and DownUp time. As shown in Figure 4 taking the double keys “WO” as an example, its characteristics are described as follows:

(1) Hold [W]: The duration of key “W” from press to release, likewise Hold [O];

(2) DD [W] [O]: The interval between press “W” (down) key to press the “O” (down);

(3) UD [W] [O]: The interval between bounce “W” key (up) to press “O” (up) key.

3.2.3. Train and Test of the Keystroke Model. For the double key characteristics in the keystroke process, the system adopts the decision tree algorithm for model training, as shown in Algorithm 1. First, Shannon entropy and information gain are selected as the criteria for feature selection of the decision tree. Second, the 7 double keys features are calculated one by one to obtain the current information gain, so as to constantly update the maximum information gain and the best features. After that, the current subtree is created according to the best feature data, and the current best feature is continuously removed to complete the recursive creation of the entire subtree. Finally, after the entire decision tree is built, the decision tree generated by training is returned. In addition, we adopt Pessimistic Error Pruning (PEP), and the penalty factor is set to 0.5 to prevent overfitting.

The adoption of the decision tree is due to the fact that once the training is completed, the distinguishing and classifying of the existing features are very fast in the testing stage. Therefore, in the process of user’s continuous keystroke in the authentication stage, keystroke data within a short period will contain 7 predefined double key characteristics with a high probability. During this time, user identity determination will be quickly completed and authentication results will be calculated through the decision tree model immediately.

Table 3: The statistical table of double keys.

No (#)	Double keys	Frequency	No (#)	Double keys	Frequency
1	AN	9619	11	WO	32
2	NG	7580	12	AO	3220
3	IN	7338	13	NA	3015
4	SH	6605	14	EI	2900
5	EN	6049	15	HE	2653
6	IA	5932	16	HS	2622
7	CH	4926	17	XI	2554
8	ZH	4145	18	ON	2466
9	AI	3869	19	HI	2337
10	JI	3798	20	IE	1906

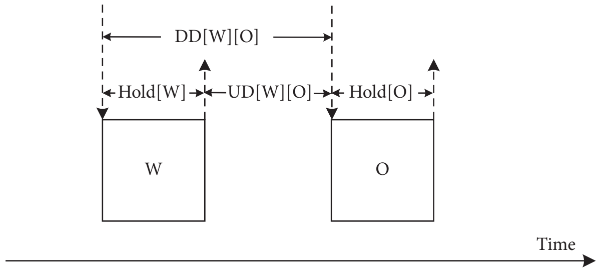

Figure 4: The time characteristics of “WO”.

3.3. Mouse Movement Recognition Model

3.3.1. Mouse Movement Data Capture. Similar to keystroke data collection, mouse movement data capture also applies hook technology, which belongs to the global mouse hook WH_MOUSE_LL in the system hook. The overall capture process includes the establishment of mouse hook, the interception and processing of mouse message, and the uninstallation of the mouse hook. The establishment and uninstallation of the mouse hook are similar to that of the keystroke hook. As for the monitoring and processing of mouse messages, it defines the unique mouse data as shown in Table 4, which is different from the keystroke data. When the user manipulates the mouse to trigger the mouse event, the mouse hook captures these messages, triggers the call back function *MouseProc*, and starts to record the mouse event type, mouse cursor coordinates (x, y), and event occurrence timestamp.

3.3.2. Mouse Movement Data Preprocessing. The original captured mouse data format is mouse event type, X coordinate, Y coordinate, and timestamp, which is relatively simple. However, mouse events have natural complexity, which can be mainly divided into four types of events: mouse idle, mouse moves, mouse drags, and mouse clicks. Among them, mouse clicks can be further divided into left click and right click, left double click, and right double click. Besides, the click events can be further divided into press (down) and release (up). Therefore, it is important to preprocess the mouse data, and transform the scattered data records into effective mouse events, and further divide them into mouse features that can be used for identity authentication.

As shown in Figure 5, the mouse data preprocessing procedure is as follows.

Step 1: mouse click events are divided into left mouse click, left mouse double click, right mouse click, and right mouse double click. The mouse hook further

```
    Input:
    The keystroke dataset matrix X;
    The keystroke feature vector F: = $F^1, F^2, \ldots, F^n$;
Output:
    The Decision Tree Model, Tree;
    initialize: do preprocess and split, data = process (X, F);
    initialize: init bestInfoGain = 0.0, bestFeature = -1
    calculate Shannon entropy; shang = calculateshang (data)
    for curFeature = 0 to n do
        calculate newEntropy and curInfoGain
        bestInfoGain = max (curInfoGain, bestInfoGain)
        bestFeature = curFeature
    end loop
    for value = 0 to data[bestFeature]. size () do
        Tree[bestFeature][value] = createTree (X, F - bestFeature)
end loop
return Tree
```

Algorithm 1: KeyStorke Model's Train [createTree].

Table 4: Mouse movement data capture.

Field	Type	Description
mouseEvent	Int	Mouse event type
mouseX	Int	Cursor x coordinate
mouseY	Int	Cursor Y coordinate
mouseStamp	Long	Mouse event timestamp

divides the left click and right click into left/right press and left/right release events, which are recorded, respectively. Therefore, the mouse click records are summarized and reformatted into the format of mouse left/right click, X coordinates, Y coordinates, press timestamp, and release timestamp.

Step 2: delete the null values. After clicking the event summary, the entire row of records with blank values in all mouse data will be deleted.

Step 3: classify the mouse data because it is difficult to extract effective features from the complicated mouse data. According to the time stamp record and pixel distance, mouse events are limited and the corresponding noncompliant data are eliminated as follows.

(1) The coordinates of the mouse cursor remain unchanged for 1 s, and the mouse is deemed to be stationary
(2) If the mouse cursor moves more than 30 pixels, it will be regarded as mouse movement
(3) If the mouse cursor changes for more than 1 s and the moving distance is greater than 30 pixels, while the left mouse button is not released when pressed, the mouse will be deemed as a drag

Step 4: define sessions to partition mouse behaviour events. The number of mouse behaviour events within a session is called session length X, and the average mouse operation time reaching session length X is called a time slice T. When the time slice is fixed, the more effective the mouse events in a session, the more stable the user behaviour characteristics and the higher the identification accuracy. However, the length of the time slice is proportional to the number of effective mouse events. The longer the time slice, the more effective the mouse events must be. However, excessively long time slice cannot guarantee the timely detection of abnormal users, which violates the original intention of the system design.

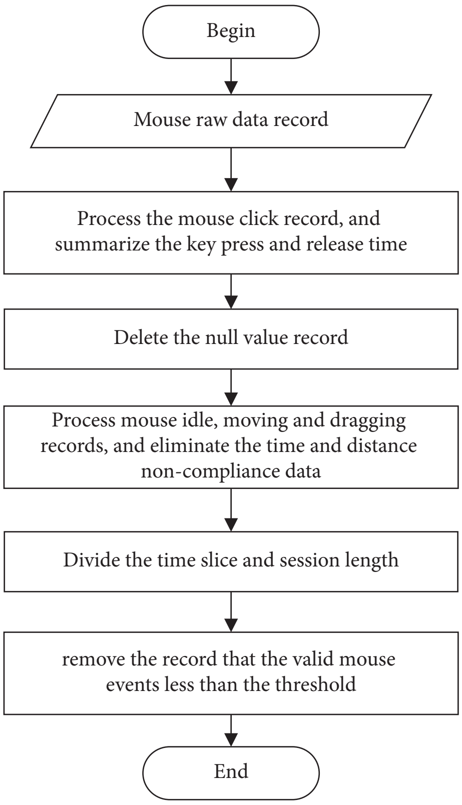
Begin
Mouse raw data record
Process the mouse click record, and summarize the key press and release time
Delete the null value record
Process mouse idle, moving and dragging records, and eliminate the time and distance non-compliance data
Divide the time slice and session length
remove the record that the valid mouse events less than the threshold
End

Figure 5: Flow of mouse data preprocessing.

Step 5: further simplify the mouse record according to the selected time slice $T$and session length X. When the number of valid mouse events in time slice T is less than X, this session event is discarded without further feature extraction.

3.3.3. Selection of Mouse Features. In the mouse recognition model, the system extracts the mouse features according to the user's mouse behaviour for verification. The mouse features are complex and diverse, and the user identity can be effectively measured by using the features of time, position, frequency, and mouse trajectory. In order to ensure the timeliness and accuracy of the model in the continuous authentication, the system chooses mouse movement with more obvious characteristics in a short period.

When the mouse cursor moves from the point P_1 (x_1, y_1) to P_2 (x_2, y_2), it shows the following five characteristics during the movement:

(1) The proportion of mouse movement events in 8 different movement directions.

(2) The moving distance (Euclidian distance) of the mouse in 8 directions including average moving distance and extreme moving distance. The calculation of the average moving distance is shown in Equation 3. The calculation of the extreme moving distance is the maximum moving distance in a single time slice T.

$$d = \frac{1}{M} \sum_{0}^{M} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{3}$$

(3) The moving speed of the mouse in 8 directions, including average moving speed and ultimate moving speed. The calculation of the average moving speed is shown in Equation 4. The calculation of the extreme moving speed is the maximum moving speed in a single time slice T.

$$v = \frac{\Delta d}{\Delta t} = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{t_2 - t_1}. \tag{4}$$

(4) The moving acceleration of the mouse in 8 directions is shown in Equation 5.

$$a = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{(t_2 - t_1)^2}. \tag{5}$$

(5) The proportion of mouse movement events in all mouse operation events.

3.3.4. Training and Testing of the Mouse Model. In the fixed time slice, the distribution of mouse behaviour events does not have regularity, so the mouse movement characteristics extracted by this system also have a small sample size and do not conform to the normal distribution. Therefore, it is difficult to obtain a good recognition effect on the classification model based on statistics or neural network. Many studies have proved that SVM performs well in small sample data and nonlinear high-dimensional mode. After comprehensively considering the number of mouse features and the samples, this paper chooses linear Support Vector Machine (SVM) [52] as the classifier of mouse recognition model, and uses the open source *libsvm* 3.0 [53] to build the classification model. Furthermore, since the gamma parameter in the Gaussian kernel will affect the width of the Gaussian function, the larger the gamma, the easier it is for the SVM to overfit. So our system sets gamma to 0.5.

Our system first uses Principal Component Analysis (PCA) for dimensionality reduction processing of the early collection of multidimensional mouse motion features, and retains the correlation of each feature to avoid the occurrence of dimensional disasters. After feature selection is completed by PCA, the original 45-dimension mouse features are reduced to 16-dimension features.

The mouse recognition model based on *libsvm* is divided into training stage and testing stage. The algorithm description is shown in Algorithm 2. Since the feature dimension reduction is required at both stages, the training and testing of the mouse model are summarized in the same function description and distinguished by option O.

X is the characteristic matrix of the mouse, which is divided into training set and test set according to different O values. L is the mouse label matrix, and the size of the matrix depends on the number of samples n. This system is a binary classification model [54], so the label is defined as (0, 1), where the legal user is 1 and the illegal user is 0. Op for *libsvm* training custom parameters, including kernel function and other values, in this system is mainly selected by using the exhaustive method.

In the training stage, the system performs dimensionality reduction on the features of the training set, conducts training according to the *libsvm* options of the feature data set, and finally exports the mouse recognition model MM after the training. In the testing stage, the system predicts according to the existing model MM and test set data, and exports the user identification result, which is legal or illegal, and calculates the classification accuracy ACC.

After that, according to the mouse data and mouse recognition model, the session length X and time slice Twere tested, in which X was 50,100, and 200 valid mouse events. The experimental data set was user mouse operations collected within 48 hours, including about 15,000 effective mouse operation events.

Figure 6 shows the ROC of session length $X = 100$. FAR is negatively correlated with FRR, and when FAR = FRR, its value is ERR. In addition, when X are 50 and 200, the ROC trend is the same as the whole, but the error rate ERR and the average time slice $T$are greatly different. As shown in Table 5, with the increase of the session length X, the ERR is reduced. This is because, the more effective the mouse events in the session cycle, the more stable the mouse features displayed by users. However, at the same time, the longer the session length is, the greater the corresponding average session time T will be, which will lead to the longer user behaviour detection time and therefore greatly affect the system's timely

```
Input:
    The mouse dataset matrix $X$;
    The option of train or test, $O$;
    The mouse label vector $L := L^1, L^2, \ldots, L^n$;
    The libsvm options $op$;
    The number of principal components, $c$;
Output:
    The Mouse Model, $MM$;
    The Predicted Answer, $Ans$;
    The Predicted Accuracy, $Acc$;
initialize: do preprocess, $pca = \mathrm{PCA}\ (c)$;
initialize: $pca$. fit $(X)$
if $O == 0$ then
    $MM = \mathrm{libsvmTrain}\ (L, X, op)$
    return Mouse Model, $MM$
else if $O == 1$ And $MM$ is exist then
    $[Ans, Acc] = \mathrm{libsvmPredict}\ (L, X, MM, op)$
    return $[Ans, Acc]$
else
    logging illegal options
end if
```

Algorithm 2: Mouse Model's Train And Test.

authentication and interception of illegal users. In conclusion, our system has made the balance among the above factors and selected the time slice length as 5 min and the session length as 100 effective mouse events.

3.4. Application Usage Recognition Model. Different from the biological behaviour feature recognition based on keystroke and mouse, the application feature recognition is based on the statistical analysis of the user's application records, and mines the user's behaviour features. When the system is deployed, it analyses the user application records in the current time window in the form of sliding window, extracts the features and standard model library for verification, and completes the authentication. Compared to the behavioural feature model, the data acquisition cycle of the application recognition model is longer, but the number of users' core applications is relatively fixed. So, the application features are more stable, which remedies the shortcoming of strong real-time but insufficient stability of keystroke and mouse recognition in multimodal recognition.

3.4.1. Application Data Collection. Application data collection mainly captures the process information, and our system adopts the Windows API PSAPI library to finish this work. When the user starts the system, the current process data are initialized through loading dynamic DLL.

First, *EnumProcesses* () enumerates the ongoing processes, counts the total number of processes, and obtains detailed data of each process (time, process ID, process name, process path) as shown in Table 6.

Second, when recording application processes data, our system adds process state, construct times, destroy times, and total running time fields according to the current process information, and initializes the value as 1, 1, 0, $t + \Delta t$.

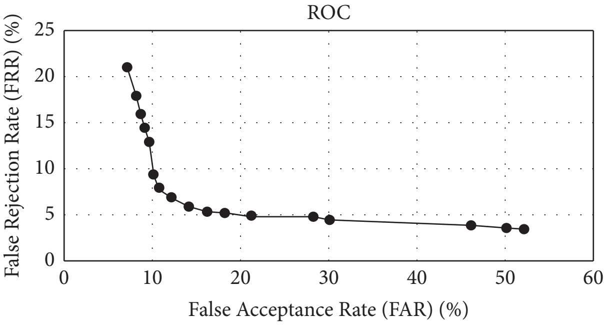

Figure 6: ROC curve graph for session length $X = 100$.

Table 5: Mouse session length and classification error rate.

Session length	Session duration (min)	ERR (%)
50	2.7	17.93
100	5.3	9.27
200	12.9	7.11

After that, our system cyclically monitors the process status, records the process construct, destroys the events, and updates the times and the total running time of the process. The specific process collection information is shown in Table 7.

3.4.2. Application Data Preprocessing and Feature Selection. The application-based identity recognition model is a statistics-based classification model, so it does not involve multidimensional features, and does not require complex dimensionality reduction and feature selection. After users log into the system, they are allowed to make basic system settings and manually select the list of applications to be monitored. Therefore, our system only carries out statistical processing for monitoring applications defined by users. First, the nontarget application information in the collected dataset will be eliminated, and then the event update frequency, running time, and total proportion of each target application are calculated. When the user does not define a monitoring application, the entire application process is handled by default.

3.4.3. Application Data Training and Test. For the training and prediction of the application recognition model, our system uses the Naive Bayes algorithm based on *sklearn naive_Bayes* library. The idea is to conduct model training according to the existing user characteristics and classification results. After the training is completed, the probabilities of each feature belonging to a different category are calculated in the testing stage as the final classification results. Therefore, it is also known as the classification algorithm based on statistics, and the related processing procedure is shown as follows.

(1) Assume that $X = \{x_1, x_2, ..., x_n\}$ is a user to be classified, and each user contains n application feature x_i

(2) The result of user identity classification is $Y = \{0, 1\}$, in which 0 means illegal user and 1 means legal user

(3) Calculate the probability $P\ (Y_i \mid x)$ that x belongs to the classification result Y, and $P\ (Y \mid x) = \text{Max}\ \{P\ (Y_1 \mid x), P\ (Y_2 \mid x))$

Algorithm 3 shows the training procedure. First, the data of the training set is normalized and transformed. Second, the Gaussian Bayesian algorithm in naive Bayes [55] is selected for model training. After the training is completed, the fitting process is carried out, and the recognition model PM is finally output. Algorithm 4 describes the testing procedure when applying the recognition model. In the testing stage, our system normalizes the test data and computes the classification result *Ans* according to the existing training model PM and the test data set X, and calculates the output confusion matrix *Acc* according to the predefined indicators.

4. MFCA System Design and Procedure

In this section, we will introduce the design methods and procedure of the MFCA system. The MFCA system introduces multimodal fusion to analyse the collected multidimensional user behaviour characteristics, performs model training according to the characteristics, and generates the trust model to achieve continuous authentication.

The MFCA system mainly consists of three parts: first, the keystroke model, mouse model, and application model obtained from training based on keystroke data, mouse data, and application record, respectively; second, the multimodal fusion technique used to merge the classification results of the three models; third, the trust model algorithm used for continuous identity authentication.

Figure 7 shows the overall design of the MFCA system, and it can be divided into training stage and testing stage, in which the training stage mainly completes the training and fusion of multidimensional behaviour models and finally generates the trust model. In the testing stage, the authentication mechanism verifies the real-time behaviour characteristics of users through the trust model and exports the current trust score. When the trust score is lower than the predefined trust threshold, the current user is judged to be an illegal user and the MFCA system will lock the device, generate alarm, and prevent the user from using the devices. When the trust score is higher than the trust threshold, the MFCA system determines that the current user's identity is legitimate, and the user can continue to use without interference and any processing.

The following parts describe the MFCA from three aspects of model training and testing, multimodal fusion mechanism, and trust algorithm design.

4.1. Model Training and Prediction. The multidimensional behaviours of network users mainly consist of keystrokes, mouse, and application usage. Our system designs the multimodal fusion mechanism to collect user behaviour data and combines them effectively, adopts the multiple classifier fusion to avoid the limitation of the single classification and improve the accuracy of the classification results and generalized capability, and finally realizes continuous authentication.

The multidimensional behaviour model (keystroke model, mouse model, and application model) mainly consists of three stages: model establishment, training, and prediction, as shown in Figure 8. When a user registers for the first time, the system will default this user as legitimate and collect the data to establish the initial model. After that, the model will constantly update and evolve with the increase of the multidimensional behaviour data. Therefore, the authentication system will continuously collect users' multidimensional characteristic data, update the model to fit the current user behaviour characteristics while verifying the identity, and improve the identity recognition accuracy.

In the training stage, once either of the models is updated, the authentication system will trigger the iterative updating of the trust model to make the model learn the characteristics of the current users and ensure the timeliness of the model. In the test phase, once a behaviour such as keystroke has enough data for feature extraction, the MFCA system will depend on these characteristics through the trust model to generate the corresponding result. The trust model will convert multiple results as the latest trust score based on their predefined proportion, and compare it with the trust threshold to determine the legitimacy of user identity, and decide whether or not to trigger alarms.

4.2. Multimodal Fusion Mechanism. Multimodal fusion (known as multi-classifier fusion) is designed to effectively combine multidimensional features for decisions, avoids the

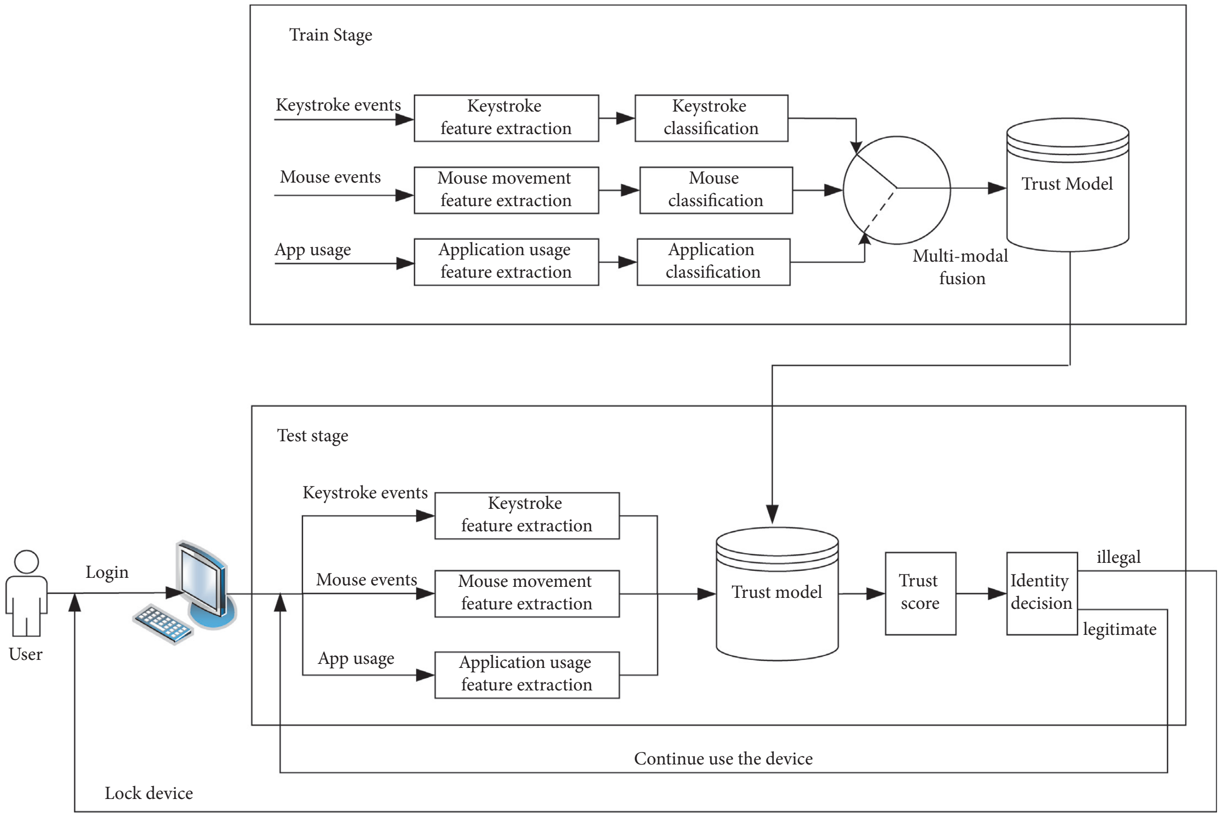

Figure 7: The overall design of the MFCA system.

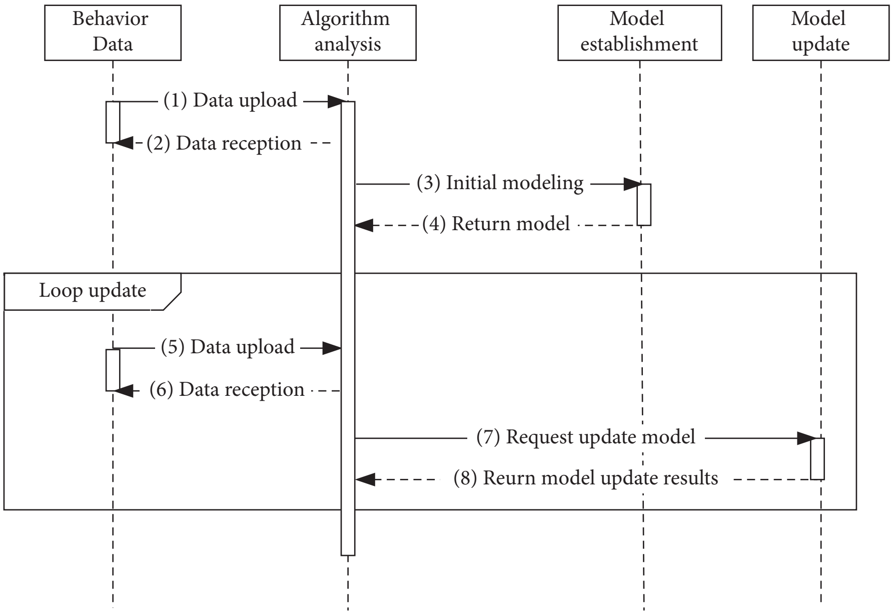

Figure 8: Model update sequence diagram.

limitations of single classification, and improves the accuracy and generalization of classification results by fusing multiple models finally. When performing continuous authentication, the complementarities among multidimensional behaviours need to be considered. In our work, three types of data, namely, keystroke, mouse movement, and application record, are collected as they have natural complementarity when users interact with the

Table 6: Initial application process information.

Time	Pid	ProcessName	ProcessPath
2019/12/11 18 : 52 : 01	10924	WeChat. exe	D:\SoftWare\WeChat\WeChat.exe
2019/12/11 18 : 52 : 01	36172	firefox. exe	D:\SoftWare\Firefox\firefox.exe
2019/12/11 18 : 52 : 01	27064	VISIO. EXE	C:\SoftWare\Visio\Office16\VISIO.EXE
……	……	……	……
2019/12/11 18 : 52 : 01	7740	KuGou. exe	D:\Download \KuGou\KuGou.exe
2019/12/11 18 : 52 : 01	6772	Microsoft. Photos. exe	C:\Program Files\WindowsApps

Table 7: Application data collect information.

Field	Type	Description
pName	String	Process name
Pid	Int	Process ID
pEvent	Int	Process event type
pStamp	Long	Process event timestamp
pPath	String	Process path
pStatus	Int	Process status
newTimes	Int	The number of process construct
delTimes	Int	The number of process destroy
totalAliveTime	Int	The duration of process

computer. Three kinds of models have different abilities to recognize users. Therefore, the MFCA can cover the using habit of different users based on multiple classifier fusion to improve the accuracy.

4.3. Trust Model Algorithm. Take behaviours of keystroke, mouse, and application as examples to describe the trust score algorithm, as shown in Algorithm 5, where the MCF adopts parallel combination, the outputs of the three base classifiers are all predefined binary values (illegal = 0 or legal = 1), and the exported results are labelled rather than probability values. The weighted voting method is selected to complete the multi-classifier fusion.

Taking keystroke identification model of double key pair "an" for example, "an" appears in the double characteristics of the weight for the W^f = Count (an)/Count (keyFeature); when a user types "an" and is identified as a legitimate user, the model exports classification results $FC = 1$. At this point, the system will reward the user with $W^f * W^K * R$, and update the trust score $\text{Trust}_i = \text{Trust}_{i-1} + W^f * W^K * R$. It should be noted that the new trust score will be no more than the maximum threshold $T_{\max}$. On the contrary, the system will punish the user with $W^f * W^K * P$ and update the trust score $\text{Trust}_i = \text{Trust}_{i-1} - W^f * W^K * P$. However, the new trust score should be no less than the minimum threshold $T_{\min}$. After obtaining the trust score Trust_i, the system will determine whether the trust score is lower than the alarm threshold T_{alert}. If the trust score is lower than the threshold T_{alert}, the system will set the warning sign Alert = 1 and trigger the alarm.

5. Performance Analysis of MFCA System

In the above, we have introduced the MFCA system and its sub-modules in detail. In this section, we will describe the experiment procedure and analyse the performance of the MFCA system in detail.

5.1. Experiment Dataset. The whole experiment scenario is free environment without static authentication, and the system does not require the user to type the specified statement to unlock the device or sign the gesture through the mouse. From data collection to authentication, the user maintains normal operations without additional restriction requirement, so that he/she can almost ignore the existence of our system except the alarm. In order to facilitate the experiment, 22 participants have been recruited to operate on computer in their daily life which can insure the continuity and integrity to reduce the impact of uncertain factors. The data are collected over three weeks after the installation of our system.

In addition, the system is applied to the general scenario rather than the strict laboratory environment. The system design considers the function and universality with the objective of balancing the application condition and the application effect, and ensures the high reliability of the characteristics selection and model training. As shown in Table 8, the computer and hardware are slightly different, but they all run on the basic Windows environment. In the keyboard and mouse equipment, the user selects the general qwerty keyboard and double key mouse. The equipment manufacturers are different, but the impact of the key feature collection of the system can be ignored.

5.2. Evaluation Metrics. After the system was deployed, 22 participants were tested to verify the performance of the MFCA system. Most previous research work adopts FAR and FRR to evaluate performance. However, it is not important to know whether an imposter or illegal user is detected, but when the illegal user is detected. In fact, FAR and FRR are more suitable for one-time authentication scenarios. They can only indicate whether an illegal user is detected but cannot indicate when an illegal user can be

```
Input:
    The process train dataset matrix $X$;
    The label vector of process data, $L := L^1, L^2, \ldots, L^n$;
Output:
    The Process Model, $PM$;
initialize: do preprocess, $scalar$ = MinMaxScaler ( )
initialize: $X$ = scalar. fit_transform ($X$)
$PM$ = GaussianNB ( )
$PM$. fit ($X$, $L$)
return $PM$
```

Algorithm 3: Process Model's Train.

```
Input:
    The label vector of process data, $L$;
    The process test dataset matrix $X$;
    The process model, $PM$
Output:
    The predicted answer, $Ans$;
    The predicted accuracy information matrix, $Acc$;
initialize: do preprocess, $scalar$ = MinMaxScaler ( )
initialize: $X$ = scalar. fit_transform ($X$)
$predicted$ = $PM$. predict ($X$)
$Ans$ = metric. classification report ($L$, $predicted$)
$Acc$ = metrics. confusion matrix ($Ans$, $predicted$, $L$)
return [$Ans$, $Acc$]
```

Algorithm 4: Process Model's Test.

detected which is more important in a continuous authentication scenario. For example, even if the recognition rate of a model is high, but the detection time is long, the intrusion may have been completed before illegal users are detected, which is unacceptable. Different from the previous performance evaluation metrics, this paper adopts Average Number of Imposter Actions (ANIA) and Average Number of Genuine Actions (ANGA) to evaluate the application effect of the system, where ANIN refers to the average number of behavioural characteristics required for illegal users to be identified as exceptions, and ANGA refers to the average number of behavioural characteristics used by legitimate users to be identified as exceptions. Therefore, ANIA should be as low as possible, so that ANIA users can be identified more quickly and in less time, which can perform fewer illegal operations. ANGA should be as high as possible so that legitimate users can work without interruption as much as possible.

5.3. Experimental Results. In the experiment, 22 participants are divided into two groups: one group comprise legal users' normal use of their own equipment, and the other group comprise illegal users' operation of others' equipment. The whole experimental environment does not have other restrictive requirements. We take the first 70% of the user's input data as training data and the others as test data. The following operations are performed on all users' input data: first, our system uses the training data of legitimate users for model training; second, the test data are used to calculate the Number of Genuine Action (NGA) of the model; finally, the data of illegal users are used to attack and the Number of Imposter Action (NIA) of the model is calculated. The initial trust score of all users is 90. When the trust score is below the threshold of 75, the pop-up alarm will be triggered, and the system will record the verification times of each feature to obtain NIA and NGA, and calculates the ANIA and ANGA. The experimental data of the two groups are shown in Tables 9 and 10.

As shown in Tables 9 and 10, ANIA = 430 and ANGA = 7341, which means that the average illegal user can be identified in the 430 features input, the legal user has an average of 7341 characteristics input. Note that an effective feature here is not a user behaviour. Take a mouse operation as an example; an effective mouse movement that contains multidimensional features such as moving distance, moving speed, and moving direction, so that the authentication speed will accelerate as the user performs the features frequently. The capture period of illegal users is shorter, which can realize the user exception in a short time. The normal using period of the legitimate user is longer; therefore, the daily work will rarely be interrupted. In addition, to speed up the abnormal authentication

```
Input:
    The feature type, $F_t$
    The feature classification results, FC;
    The weight of this feature in its recognition model, $W^f$;
    The trust score after last calculation, $\text{Trust}_{i-1}$;
    The initial trust score $T$, and score threshold $T_{\max}$ and $T_{\min}$;
    The score will trigger system alert $T_{alert}$;
    The reward and punishment score for each feature, $R$, $P$;
    The weight of Three authentication model, $W{:}\ = W^k, W^m, W^p$;
Output:
    The trust score after this calculation, $\text{Trust}_i$;
    IF alert the system trust, $A$;
initialize: Init $A = 0$, if the first calculation, $\text{Trust}_i = T$;
if $F^t = = 0$ then
    if $FC == 0$ then
        $\text{Trust}_i = \max(\text{Trust}_{i-1} - W^f * W^k * P, T_{\min})$;
    else $\{FC == 1\}$
        $\text{Trust}_i = \min(\text{Trust}_{i-1} + W^f * W^k * R, T_{\max})$;
    end if
else if $F^t == 1$ then
    replace $W^k$ in the above formula with $W^m$;
else $\{F^t == 2\}$
    replace $W^k$ in the above formula with $W^p$;
end if
if $\text{Trust}_i < T_{\text{alert}}$ then
    $A = 1$;
end if
return $[\text{Trust}_i, A]$
```

Algorithm 5: Trust Score Calculation.

Table 8: Summary of experiment setting.

Number of participants	15	7
Device types	PC	Notebook
OS	Win7	Win10
Resolution	1440∗900	1920∗1080

Table 9: Illegal user authentication NIA results.

User ID	Keystroke	Mouse	Applicate	NIA
1	124	96	43	263
2	87	157	37	281
3	141	133	43	317
4	180	121	59	360
5	209	117	48	374
6	155	213	51	419
7	281	106	79	466
8	142	231	86	471
9	189	201	97	487
10	143	239	96	521
11	279	385	115	779

Table 10: Legal user authentication NGA results.

User ID	Keystroke	Mouse	Applicate	NGA
1	1009	3042	586	4637
2	1459	3533	589	5581
3	1356	3687	901	5944
4	2217	3791	524	6532
5	2699	3664	627	6990
6	4231	2070	1138	7439
7	3715	2984	872	7571
8	5357	2158	440	7955
9	5751	1752	746	8249
10	4970	3941	656	9567
11	6628	2599	1066	10293

speed or prevent the user from being disturbed, our system can increase or reduce the trust threshold of the trust model.

6. Conclusion

This paper proposes a continuous authentication system based on multidimensional behaviour characteristics, which introduces the trust value that is changed in real-time with the user behaviour characteristics. Only when the trust score is lower than the predefined trust threshold, the current user is considered to be an illegal user and the alarm is triggered. This system fully considers the instability of biological characteristics, avoiding the nonblack and white decision of single extreme characteristics, and improving the use of real users without the relaxation of the abnormal user. In addition, in the calculation of the trust value, the system is based on the accuracy of the multiple classification models, and the reliability of the calculation can be guaranteed.

The MFCA system has the advantages of low cost and user-friendliness because of no additional hardware equipment and no additional users' operations. Therefore, the MFCA system is important for the realization of a continuous user authentication system and especially suitable for office environments with high security requirements such as finance corporations and online examination. The adoption of the MFCA can prevent the insider attacks and support the zero trust architecture. However, how to determine the trust threshold and improve performance need to be considered. Besides, the score-level fusion mechanism introduces additional calculation time, which will increase the fusion time. So, the other fusion mechanism such as rank-level fusion mechanism will be considered in our future work.

Besides, in order to apply our model in real-life scenarios, we must consider the problem of user data privacy protection. A privacy attack on a machine-learning model may expose personal information. For example, the attacker may obtain the user's mouse movement characteristics, keystroke characteristics, and application usage characteristics by attacking our model and infer the user's private information such as login passwords, private letters by analysing the characteristics in a certain period of time. In future work, we will consider analysing possible attack scenarios against the models and introduce data anonymisation and differential privacy mechanisms to protect user data privacy.

Data Availability

The source data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after the publication of this article, will be considered by the corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key Research and Development Program (2018YFE0206800) and in part by the project under contract No. 2019-JCJQ-ZD-182-00-02.

References

[1] S. Yao, J. Guan, Y. Wu, K. Xu, and M. Xu, "Toward secure and lightweight Access authentication in SAGINs," *IEEE Wireless Communications*, vol. 27, no. 6, pp. 75–81, 2020.

[2] I. Stylios, S. Kokolakis, O. Thanou, and S. Chatzis, "Behavioral biometrics & continuous user authentication on mobile devices: a survey," *Information Fusion*, vol. 66, pp. 76–99, 2021.

[3] R. Mehra, A. Meshram, and B. R. Chandavarkar, "Remote user authentication and issues: a survey," in *Proceedings of the 2020 11th, International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, Kharagpur, India, July 2020.

[4] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: a contemporary survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65–84, 2020.

[5] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[6] "Insider threat report [EB/OL]," 2020, https://www.cybersecurity-insiders.com/wp-content/uploads/2019/11/2020-Insider-Threat-Report-Gurucul.pdf,%202021-05.

[7] S. Teerakanok, T. Uehara, and A. Inomata, "Migrating to zero trust architecture: reviews and challenges," *Security and Communication Networks*, vol. 2021, Article ID 9947347, 10 pages, 2021.

[8] R. Spillane, "Keyboard apparatus for personal identification," *IBM Technical Disclosure Bulletin*, vol. 173346 pages, 1975.

[9] G. E. Forsen, M. R. Nelson, and R. J. J. Staron, "Personal attributes authentication techniques," Pattern Analysis and Recognition Corp, Technology Report, NTIS No. 197805, 1977.

[10] R. Gaines, W. Lisowski, and S. Press, "Authentication by keystroke timing: some preliminary results," Rand Corporation: Rand Report R-2560-NSF, The Rand Corporation, Santa Monica, CA, USA, 1980.

[11] R. A. J. Everitt and P. W. Mcowan, "Java-based Internet biometric authentication system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1166–1172, 2003.

[12] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system using angle-based mouse movement biometrics," *ACM Transactions on Information and System Security*, vol. 18, no. 3, pp. 1–27, 2016.

[13] U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa, "Continuous authentication of smartphones based on application usage," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 3, pp. 165–180, 2019.

[14] M. Ehatisham-Ul-Haq, M. Awais Azam, U. Naeem, Y. Amin, and J. Loo, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018.

[15] R. Kumar, V. Phoha, and R. Raina, "Authenticating users through their arm movement patterns," 2016, http://arxiv.org/abs/1603.02211.

[16] Y. Li, "Research on gesture recognition model and its application based on wear sensing perception," pp. 1–45, Lanzhou University, Lanzhou, China, 2019, Master's Thesis.

[17] J. Handa, S. Singh, and S. Saraswat, "A comparative study of mouse and keystroke based authentication," in *Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 670–674, Noida, India, January 2019.

[18] P. H. Pisani, A. C. Lorena, and P. L. F. De Carvalho, "Adaptive biometric systems using ensembles," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 19–28, 2018.

[19] M. Liu, "Research on authentication technology based on user keystroke behavior," pp. 1–80, Beijing University of Posts and Telecommunications, Beijing, China, 2019, Master's Thesis.

[20] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Transactions on Information and System Security*, vol. 8, no. 3, pp. 312–347, 2005.

[21] P. Dowland, H. Singh, and S. Furnell, "A preliminary investigation of user authentication using continuous keystroke analysis," in *Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security*, Las Vegas, NV, USA, September 2001.

[22] T. Shimshon, R. Moskovitch, L. Rokach, and Y. Elovici, "Continuous verification using keystroke dynamics," in *Proceedings of the 2010 International Conference on Computational Intelligence and Security*, pp. 411–415, Naning, China, December 2010.

[23] M. Rybnik, M. Tabedzki, M. Adamski, and K. Saeed, "An exploration of keystroke dynamics authentication using nonfixed text of various length," in *Proceedings of the International Conference on Biometrics and Kansei Engineering*, pp. 245–250, Tokyo, Japan, July 2013.

[24] X. Song, P. Zhao, M. Wang, and C. Yan, "A continuous identity verification method based on free-text keystroke dynamics," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 206–210, Budapest, Hungary, October 2016.

[25] J. Huang, D. Hou, and S. Schuckers, "A practical evaluation of free-text keystroke dynamics," in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pp. 1–8, New Delhi, India, February 2017.

[26] B. Ayotte, M. K. Banavar, D. Hou, and S. Schuckers, "Fast and accurate continuous user authentication by fusion of instance-based, free-text keystroke dynamics," in *Proceedings of the International Conference of the Biometrics Special Interest Group*, pp. 1–6, Darmstadt, Germany, September 2019.

[27] M. Pusara and C. E. Brodley, "User Re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 1–8, Association for Computing Machinery, New York, NY, USA, October 2004.

[28] Y. Aksari and H. Artuner, "Active authentication by mouse movements," in *Proceedings of the 24th International Symposium on Computer and Information Sciences*, pp. 571–574, Suzelyurt, Cyprus, September 2009.

[29] B. Sayed, I. Traore, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.

[30] C. Chao Shen, Z. Zhongmin Cai, X. Xiaohong Guan, Y. Du, and R. A. Maxion, "User authentication through mouse dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 16–30, 2013.

[31] C. Shen, Z. Cai, and X. Guan, "Continuous authentication for mouse dynamics: a pattern-growth approach," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pp. 1–12, Boston, MA, USA, June 2012.

[32] E. Medvet, A. Bartoli, F. Boem, and F. Tarlao, "Continuous and non-intrusive reauthentication of web sessions based on mouse dynamics," in *Proceedings of the 9th International Conference on Availability, Reliability and Security*, pp. 166–171, Fribourg, Switzerland, September 2014.

[33] B. Li, W. Wang, Y. Gao, V. Phota, and Z. Jin, "Hand in motion: enhanced authentication through wrist and mouse movement," in *Proceedings of the IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, Rendondo Beach, CA, USA, October 2018.

[34] M. Yildirim and E. Anarim, "Session-based user authentication via mouse dynamics," in *Proceedings of the 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, Sivas, Turkey, April 2019.

[35] Á Fülöp, L. Kovács, T. Kurics, and E. Windhager-Pokol, "Balabit mouse dynamics challenge data set," Available at: https://github.com/balabit/Mouse-Dynamics-Challenge, 2016.

[36] S. Liu, "Research on user behaviour analysis model based on log data," pp. 1–63, Yunnan University, Kunming, China, 2017, Master's Thesis of.

[37] W. Meng, Y. Wang, D. S. Wong, S. Wen, and Y. Xiang, "TouchWB: touch behavioral user authentication based on

web browsing on smartphones," *Journal of Network and Computer Applications*, vol. 117, pp. 1–9, 2018.

[38] J. Wei, "Analysis and research of user access behavior based on DNS log," pp. 1–75, Beijing Jiaotong University, Beijing, China, 2019, Master's Thesis of.

[39] A. Alzubaidi, S. Roy, and J. Kalita, "A data reduction scheme for active authentication of legitimate smartphone owner using informative apps ranking," *Digital Communications and Networks*, vol. 5, no. 4, pp. 205–213, 2019.

[40] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15274–15278, 2009.

[41] S. K. S. Modak and V. K. Jha, "Multibiometric fusion strategy and its applications: a review," *Information Fusion*, vol. 49, pp. 174–204, 2019.

[42] I. Traore, I. Woungang, M. S. Obaidat, N. Youssef, and L. Iris, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," in *Proceedings of the Fourth International Conference on Digital Home*, pp. 138–145, IEEE Computer Society, Guangzhou, China, November 2012.

[43] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77–89, 2014.

[44] L. Fridman, A. Stolerman, S. Acharya et al., "Multi-modal decision fusion for continuous authentication," *Computers & Electrical Engineering*, vol. 41, pp. 142–156, 2015.

[45] S. Mondal and P. Bours, "Combining keystroke and mouse dynamics for continuous user authentication and identification," in *Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis*, pp. 1–8, ISBA, Sendai, Japan, February 2016.

[46] I. D. S. Beserra, L. Camara, and M. D. Costa-Abreu, "Using keystroke and mouse dynamics for user identification in the online collaborative game league of legends," in *Proceedings of the 7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, November 2018.

[47] S. M. Sergio, R. S. Baker, O. C. Santos, and J. González-Boticario, "A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios," *IEEE Access*, vol. 6, pp. 39154–39179, 2018.

[48] K. Quintal, B. Kantarci, M. Erol-Kantarci, A. Malton, and A. Walenstein, "Contextual, behavioral, and biometric signatures for continuous authentication," *IEEE Internet Computing*, vol. 23, no. 5, pp. 18–28, 2019.

[49] R. Wang and D. Tao, "Implicit authentication mechanism based on context awareness for smartphone," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 6, pp. 118–125, 2019.

[50] Y. Yang, J. Sun, and L. Guo, "PersonaIA: a lightweight implicit authentication system based on customized user behavior selection," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 113–126, 2019.

[51] S. Vhaduri and C. Poellabauer, "Multi-modal biometric-based implicit authentication of wearable device users," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3116–3125, 2019.

[52] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[53] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[54] S. Almalki, P. Chatterjee, and K. Roy, "Continuous authentication using mouse clickstream data analysis," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 76–85, Springer, Atlanta, GA, USA, July 2019.

[55] I. Rish, "An empirical study of the naive Bayes classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41–46, 2001.