An Efficient Man-Machine Recognition Method Based On Mouse Trajectory Feature De-redundancy

Xiaofeng Lu
luxf@bupt.edu.cn
School of Cyberspace Security, Beijing University of Posts and Telecommunications
Beijing, China

Zhenhan Feng
pixel@bupt.edu.cn
School of Cyberspace Security, Beijing University of Posts and Telecommunications
Beijing, China

Jupeng Xia
jupeng.xia@antgroup.com
Alipay (Hangzhou) Information & Technology Co., Ltd.
Hangzhou, China

ABSTRACT

Behavioral authentication codes are widely used to resist abnormal network traffic. Mouse sliding behavior as an authentication method has the characteristics of less private information and easy data sampling. This paper analyses the attack mode of the machine sliding track data, extracts the physical quantity characteristics of the sliding path. Features importance scores are used to select the candidate features, and further Pearson correlation co- efficient is used to filter out the features with high correlation. This paper use XGBoost model as a classifier. In addition, an efficient evasion attack detection method is proposed to deal with complex human behavior evasion attacks. The experiment was carried out on two mouse sliding datasets. The experimental results show that the proposed method achieves 99.09% accuracy and 99.88% recall rate, and can complete the man-machine identification in 2ms.

CCS CONCEPTS

• **Security and privacy → Biometrics**; • **Computing methodologies → Boosting**.

KEYWORDS

Behavior Authentication Code, Human-Machine Recognition, Mouse Trajectory, Evasion Attack Detection, Machine Learning

ACM Reference Format:
Xiaofeng Lu, Zhenhan Feng, and Jupeng Xia. 2021. An Efficient Man-Machine Recognition Method Based On Mouse Trajectory Feature De-redundancy. In *Annual Computer Security Applications Conference (ACSAC '21), December 6–10, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485832.3485895

1 INTRODUCTION

Malicious robots programs register on some websites to earn financial returns such as coupons offered by websites that cause hundreds of millions of sites to lose.Human-machine identification (HMI) has been used as a verification method to resist the traffic of automated robots. HMI requires users to complete specified operations, such as input verification codes, graphics verification codes and voice verification codes. The system identifies whether the user completing the operation is a human or a robot program. Due to the rapid development of image recognition technology, traditional graphics verification code technology is easy to crack, and has been gradually abandoned because of poor user experience. In essence, slider type verification is used for human-machine identification by using biological behavior characteristics, which is used in the scene of registering new users, user login, preventing malicious operation and data anti-crawlers. The recognition of machine behavior in slider verification is not only whether the jigsaw puzzle can be completed, but also analyzes the behavior trajectory data generated by human users or robot programs, so as to quickly and accurately return the results of human-machine judgment. Sliding verification based on machine learning improves the attack cost and plays a key role in dealing with the generation of malicious machine traffic[2].

Web-based automation technology is developing rapidly. In addition to user-defined automation technology through programming, capture-replay technology on the web has become the focus of research. Replay technology can be used to complete complex and repetitive tasks, thereby reducing the workload of developers and improving development efficiency[17]. Attacker programming custom sliding scheme is the machine sliding behavior in this paper. Attacker uses capture-replay technology to slide slider. Because it is a machine replaying human behavior, the method of machine learning alone is not enough to identify. Therefore, an evasion attack detection method is proposed, which combines with machine learning method to achieve more accurate human-machine identification results.

Human-machine identification should have the features of fast recognition, simple operation and high accuracy. This requires that the machine learning model should be reliable, robust, causal, scalable and universal, and it can control the input changes caused by disturbance to affect the performance of the model. Finally, as human-machine identification is applied in different scenarios, the machine learning model should be highly portable[7].

This paper presents a mouse sliding track classification technique based on feature de-redundancy and XGBoost algorithm. In this paper, the mouse sliding track is studied to extract the relevant features of human behavior and robot program behavior. The structure fractional gain method is used to filter out the important features, and Pearson correlation coefficients between the features

ACSAC '21, December 6–10, 2021, Virtual Event, USA
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8579-4/21/12…$15.00
https://doi.org/10.1145/3485832.3485895

are calculated to filter out the independent features. Finally, XGBoost model algorithm is used to learn features, and we can get a more accurate and robust model.

The main contributions of this method are as follows:

(1) In terms of valid features, instead of directly using trajectory features from other literatures, we used feature derivation and de-redundancy methods, and selected two features obtained from attack model study that are more effective in classifying machine behavior, which make the trained models have better generalization ability.

(2) Machine behavior based on behavior replay can avoid model detection easily. We use behavior similarity based detection method to detect replay behavior, and change the algorithm process to make it perform better in time efficiency.

2 RELATED WORK

Human-machine validation based on slider-type validation codes has been widely used in recent years. Machine behavior has evolved from simple sliding to approximate human behavior, and the recognition model has been optimized and improved. Machine learning classification after extracting the sliding characteristics of human and machine is the main research direction at present. Kang[9] uses the method of feature group hierarchy and uses random forest model training features to classify tracks. After feature extraction, Zhang[21] describes in detail the construction process of the decision tree for gradient promotion and the method of classification learning, and compares the training effects of other models. OuYang[13] mainly merges the XGBoost model with the lightGBM model with the highest confidence level sample as the prediction result. Xu[20] also used the random forest method to train the model on a small sample set and achieved excellent results. The above methods of human-machine identification only include the comparison between human behavior and machine behavior. The typical feature analysis of machine behavior is insufficient. The validity of each feature was not considered in feature selection, for example, whether there is a positive impact on the classification results, or whether there is feature redundancy. Experiments on a single dataset are not enough to verify the model performance also.

Solano[17] proposed the replay attack behavior in the article and set up a replay attack algorithm to attack the machine learning model. The attack success rate is as high as 87.3%. The algorithm generates a new track by calculating a certain offset of the track, which is a slide without direction restriction.We name this replay attack as evasion attack.In this paper, we also consider the trajectory in the case of pixel deviation. Unlike Solano[17], the human-machine identification glide is a directional glide, and the end points must match before a complete trajectory can be calculated. In the evasion attack detection experiments, we consider the deviation and discuss the influence of detection algorithm and deviation.

Similar to the human-machine recognition technology based on mouse slider verification, mouse slide dynamics can also be used for identity authentication. Due to the different habits of different people operating the mouse, the behavior of different mouse operators can be distinguished. Ma[11] et al. proposed a user authentication method based on mouse-click behavior on a soft keyboard, which models an unfixed mouse track and uses SVM and majority voting to authenticate users. Shen[14] et al. proposed using kernel PCA to map raw data to high-dimensional space using a non-linear mapping function to effectively solve behavioral variability. Antal[4] uses Balabit public datasets to train motion-specific classifiers, and uses two scenarios to evaluate sliding, clicking, and dragging behaviors, respectively. Studies show that dragging behavior is better for classifying specific actions. Jorgensen[8] et al. evaluated the authentication method based on mouse dynamics, performed experiments on controlled environments, remote access scenarios, and the impact of classifiers on the results, pointing out the drawbacks of using this method, such as inadequate control of environmental variables and too long verification time. Mouse dynamics-based identity authentication method is not suitable for commercial platforms with high access because of its cumbersome operation and long authentication time.

3 MACHINE ATTACK MODEL

3.1 Direct Machine Behavior

Five direct attack modes were obtained by classifying the mouse behavior of the robot program in the training dataset. Each of the following plots is an X-Y and Time-X coordinate of the two sample data.

(1) First Attack Mode

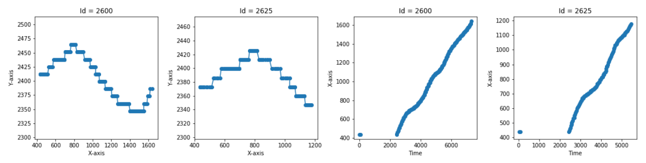

Figure 1: X-Y and Time-X of the first attack mode.

In the first attack mode, the robot program mimics the human longitudinal offset and skips to another vertical level to continue sliding after a short distance of lateral sliding. From the whole track, there is no difference between the machine behavior and the human behavior, but from the time displacement coordinates, we find that there is no significant acceleration and deceleration of the robot program behavior, and the sampling points are dense. Time-X coordinates show that the mouse stays at the starting point for a long time. Combined with the feature of slider verification code, it is not difficult to deduce that the robot program clicks on the starting point of the slider to get the shadow part location. Then the length of sliding is calculated by image recognition or other methods, and the sliding length is divided into several coordinate points before sliding. Because the track collection process is from the moment the slider is pressed, no sliding behavior occurs during this period of time when the program first calculates the end point of the slide, and a larger time difference occurs between two points of normal human behavior.

(2) Second Attack Mode

In the second attack mode, similar to the first attack mode, the program activates the shadow part by clicking the slider, which also takes time to calculate the shadow position and sliding distance. By observing the Time-x diagram, we can see that the robot program imitated human acceleration and deceleration behavior to complete

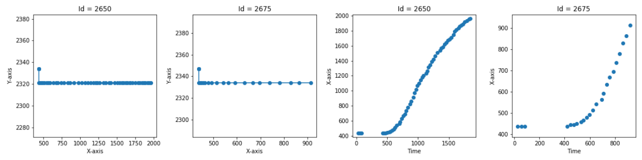

Figure 2: X-Y and Time-X of the second attack mode.

the sliding at an extremely fast speed. By calculating the sample data, we find that the acceleration time period generated by the program behavior is much larger than the deceleration time period, but the deceleration time period generated by human behavior is larger than the acceleration time period. This feature is expressed by numerical value, that is, the average value of the differential speed as the distinguishing feature. From the trajectory point of view, in the current attack mode, machine behavior produces little vertical deviation but only lateral sliding. Therefore, the percentage of vertically invariant path points over the entire sliding path is proposed as the trajectory feature.

(3) Third Attack Mode

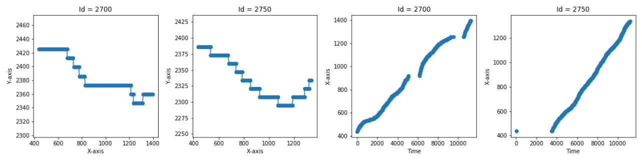

Figure 3: X-Y and Time-X of the third attack mode.

In the third attack mode, instead of spending a lot of time clicking on the slider and activating the shadowed part, the robot program moves at a near constant speed throughout the sliding process, which is a shorter random jump motion from the slider track. Differential displacement characteristics can play a key role in the behavior where the distance between two points decreases when the sample points are dense. When investigating current attack patterns, we found that there are a large number of identical robot program behavior tracks, and feature similarity detection is also applicable to machine behavior.

(4) Fourth Attack Mode

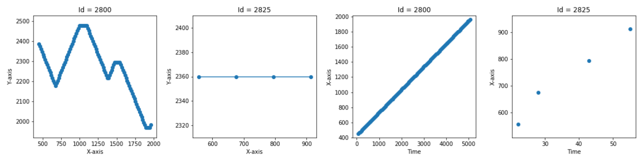

Figure 4: X-Y and Time-X of the fourth attack mode.

The fourth attack mode is an approximate uniform motion with vertical deviation, moving to the next coordinate point every same time interval, and the time interval is nearly equal due to sampling delay. When calculating the variance of difference time, the numerical values of human behavior fluctuate widely. The variance of machine behavior of the fourth attack mode is minimum or even 0. The current attack mode is easy to distinguish human behavior.

(5) Fifth attack mode

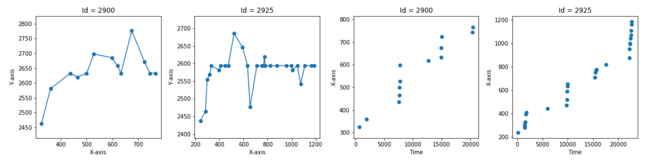

Figure 5: X-Y and Time-X of the fifth attack mode.

In the fifth attack mode, from the X-Y displacement diagram, we can see that the trajectory of the machine behavior is irregular and random, and the trajectory is chaotic. Under the time characteristic sequence, there are multiple acceleration and deceleration motions and abnormal acceleration phenomena in the machine behavior. There is a large time span between the two adjacent acceleration stages. Compared with human acceleration behavior, the robot program can complete the acceleration quickly.

3.2 Evasion Attack

Mouse sliding evasion attacks require the establishment of attack databases, which contain sliding tracks of different lengths that are samples of real human mouse sliding behavior. In real environment, the mouse sliding evasion attack uses image gray difference to identify the location of the gap, records the center point or edge of the gap as the end coordinate, calculates the distance difference between the start and end coordinates, finds a sliding sample of length in the attack library, and calculates the new coordinate point and delay time. The track of an evasion attack resembles a fast jump behavior, jumping from one current coordinate point to another or not moving after a delay in interval.

Since the trajectory of evasion attack originates from real human behavior, it is difficult for traditional machine learning feature model methods to detect the evasion attack correctly.

When analyzing the sample set, we found not only human behavior evasion attacks, but also identical or similar tracks in machine behavior. When evasion attacks are performed with automated tools such as Selenium or Quick Macro, the tracks may not be identical due to time bias or sampling point collection bias, but evasion attacks can be detected from the perspective of feature similarity. Therefore, evasion detection of all sample sets can be a prerequisite for model prediction.

4 HUMAN SLIDING BEHAVIOR

We processed the human mouse sliding data graphically and sampled the human sliding sample data equidistantly. After drawing the X-Y diagram of the moving path, we can see that the human behavior is similar to the machine behavior in the sliding trajectory by observing the coordinate diagram.

Machine behavior can be divided into simple machine behavior and complex machine behavior. Simple machine behavior focuses on the starting and ending positions, with little or no consideration for sliding modes. Hopping movements with little or no lateral deviation are common. Complex machine behavior mimics human

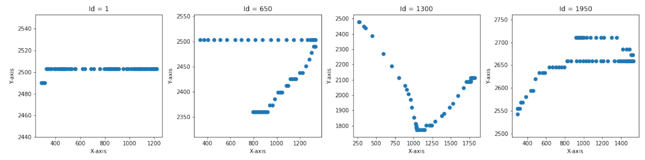

Figure 6: X-Y of human behavior.

acceleration, deceleration, and bias, but there are few reentry phenomena. From T-X figure of machine behavior and human behavior, we can more clearly observe the changes of mouse point displacement under time conditions and derive velocity characteristics.

human.png

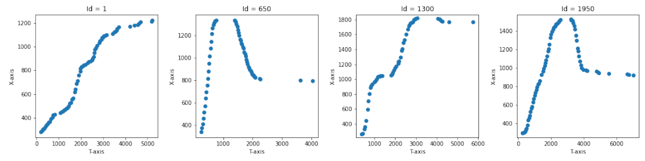

Figure 7: Time-X of human behavior.

By looking at the time series coordinate of Figure 7, we found that there is a significant difference between human and machine sliding behavior in time series. Machine behavior was more similar to uniform motion, and the sample points were arranged densely. Mouse sliding track is time series data, we use fuzzy entropy to distinguish the fuzziness of series data, The formula for calculating fuzzy entropy is[5]:

(1) Defines the distance d_{ij}^n between two n-dimensional vectors X_i^n and X_j^n to be the largest difference between their corresponding elements, Defines the similarity D_{ij}^n of two vectors X_i^n and X_j^n with the fuzzy function $\mu(d_{ij}^n, m, r)$, that is:

$$D_{ij}^n = \mu(d_{ij}^n, m, r) = e^{\frac{(-d_{ij}^n)^m}{r}} \tag{1}$$

In the formula (1), $\mu(d_{ij}^n, m, r)$ is an exponential function, where m and r are the gradient and width of the exponential function boundary, respectively.

(2) Define function

$$O^n(m,r) = \frac{1}{M-n} \sum_{i=1}^{M-n} \{\frac{1}{M-n-1} \sum_{j=1, j\neq i}^{M-n} D_{ij}^n\} \tag{2}$$

(3) Fuzzy entropy is defined as

$$FuzzyEn(n,m,r) = \lim_{M\to\infty} [\ln O^n(m,r) - \ln O^{n+1}(m,r)] \tag{3}$$

Because the sampling intervals were different, the displacement coordinates cannot be used as the basis for judging the fuzzy entropy. We used the velocity values of each small segment of displacement as the input sample series of the fuzzy entropy. Figure 8 shows the fuzzy entropy of 3000 sample data. Of the 3000 samples, the first 2599 were human behavior, and the 2600 to 3000 were machine behavior. Figure 8 shows that the fuzzy entropy value of simple machine behavior is low, the speed variation value is regular,

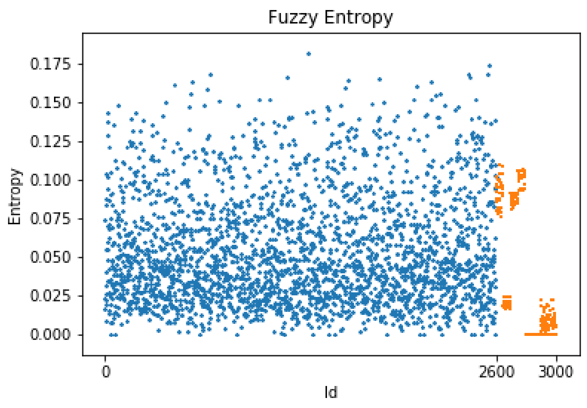

Figure 8: Fuzzy entropy.

and can be easily distinguished from human behavior. Complex machine behavior mimics human motion track acceleration and deceleration, and the speed sequence changes greatly, so the fuzzy entropy value remains stable at a high level. The uncertainty of human sliding track speed is high, but there are multiple uniform motion segments in the track when fast sliding occurs.

From the point of view of feature extraction, the calculation of fuzzy entropy value is extremely time-consuming, and it takes about 130 seconds to calculate 3000 samples, which does not meet the requirements of fast validation.

5 A FAST HUMAN-MACHINE RECOGNITION METHOD BASED ON MOUSE SLIDING

5.1 Feature Extraction

Mouse gliding behavior as a motion track has motion-related physical quantities, the mouse itself is considered as a particle, and the sliding plane is used as a two-dimensional reference system to obtain parameters related to displacement, speed and acceleration. In human recognition verification, a reference coordinate system is established with the upper left corner of the screen as the origin, and track data is collected when the slider verification code is sliding horizontally. Human behavior has longitudinal deviations during lateral gliding, with significant acceleration and deceleration, stationary and retraction phenomena. Simple machine behavior jumps significantly, including sampling points with large displacement spans, and complex machine behavior mimics human gliding to produce longitudinal deviation and static behavior. We calculate the sliding characteristics from displacement, speed and acceleration information, and extract additional feature information based on other differences between human and machine tracks.

Defines the difference distance as the value of the last point in the whole distance vector minus the value of the previous point. Similarly, the differential velocity and differential acceleration can be obtained. The time vector is the desensitization time stamp collected when the coordinate point changes. We used different datasets, and the sampling frequencies are different. So, the time information is not used as a feature. However, the time difference is used as the basis for classification.

Define the horizontal coordinate vectors separately $\vec{X} = [x_1, x_2, x_3, \ldots, x_n]$, vertical coordinate vector $\vec{Y} = [y_1, y_2, y_3, \ldots, y_n]$, time vector $\vec{T} = [t_1, t_2, t_3, \ldots, t_n]$. Coordinate difference can be computed, $\vec{\Delta X} = [\Delta x_1, \Delta x_2 \Delta x_3, \ldots, \Delta x_{n-1}]$, $\vec{\Delta Y} = [\Delta y_1, \Delta y_2 \Delta y_3, \ldots, \Delta y_{n-1}]$, time difference vector $\vec{\Delta T} = [\Delta t_1, \Delta t_2 \Delta t_3, \ldots, \Delta t_{n-1}]$, velocity vector $\vec{V} = \frac{\vec{\Delta X}}{\vec{\Delta T}} = [v_1, v_2, v_3, \ldots, v_{n-2}]$, acceleration vector $\vec{A} = \frac{\vec{V}}{\vec{\Delta T}} = [a_1, a_2, a_3, ..., a_{n-2}]$. Velocity difference vector can be obtained by the same calculation $\vec{\Delta V} = [\Delta v_1, \Delta v_2 \Delta v_3, \ldots, \Delta v_{n-2}]$ and acceleration difference vector $\vec{\Delta A} = [\Delta a_1, \Delta a_2 \Delta a_3, \ldots, \Delta a_{n-2}]$.

We divide the entire sliding track into three stages, the start segment, the middle segment and the end segment. For each physical quantity, start, middle, and end points can be calculated, and machine behavior and human behavior differ from each other in local feature analysis. Meanwhile, maximum, minimum, mean and standard deviation are used to describe the feature information of the whole sliding process.

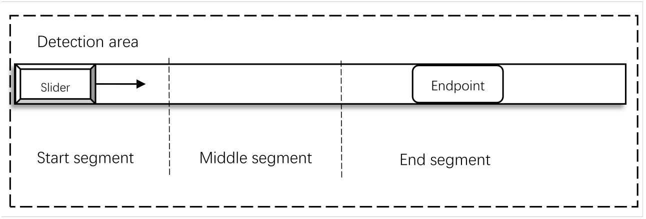

Figure 9: Example diagram of slider verification codes.

As shown in Figure 9, due to the large or small longitudinal deviation when human operates mouse movement, Therefore, from the perspective of two-dimensional space, the detection area is not only the slider track area. The lateral and vertical features are useful for human-machine classification, such as the lateral differential velocity and the vertical differential velocity, so the lateral and vertical sliding features in two-dimensional space are calculated separately. Because the slider verification code itself is a lateral directional sliding behavior and contains only minimal vertical deviation. Therefore, from the point of view of feature distinction, the vertical vector feature distinction is low, while the horizontal vector feature distinction is high.

In addition to the above calculated features, we also extract some other features, such as Euclidean distance, the sum of differential distances, the distance between maximum transverse coordinate and target point, the relationship between maximum transverse coordinate and final point, which is used to judge the retrace of sliding, and the longitudinal deviation which is used to judge the stability of sliding.

After the above feature extraction methods, 119 feature vectors are obtained.

5.2 Feature Selection

XGBoost calculates and selects which feature as the split point based on the increase of the structure fraction. The importance of a feature is the sum of the number of times it occurs in all trees. The more occurrences of a feature used to construct a decision tree in the model, the more important the feature will be. To avoid feature redundancy and noise problems caused by too many features, 20 features with scores higher than 10 were selected as candidate features after model fitting training.

In addition, the problem of collinearity exists even when features with high scores. Feature collinearity and high correlation can bias the trained model. The specific manifestations are as follows:

(1) The feature importance of strong features decreases. If there are many groups of strong features with high correlation, the feature importance of strong features will be diluted and the validity of features will be reduced.

(2) It affects the generalization performance of the model, and there are a number of highly correlated features in the mouse sliding behavior, If the vertical deviation is small, the displacement, velocity and acceleration information will not change. It makes displacement and velocity, which should be irrelevant, appear collinearity. One of the reasons for the good generalization performance of XGBoost is that the base models in different feature spaces and sample spaces can be constructed by row-column sampling. If there is a large amount of collinearity in the features, the types of subspaces that can be used in the base model will be reduced, the integration model will be weakened, and the diversity will be reduced, so the generalization performance of the model will also be reduced.

(3) High-dimensional features consume too much time and memory, and produce noise (insignificant features) interference. Although XGBoost is robust to noise in a large sample set, splitting trees like XGBoost on irrelevant features will not properly assess the validity of features.

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{4}$$

There is a linear correlation between the characteristics of the mouse slide track data, for example, because the lateral slide span is large and there is only a slight deviation in the longitudinal slide, there is a linear correlation between the lateral velocity or distance and the two points (coordinate points of the two-dimensional plane).

Pearson correlation coefficient can effectively assess the correlation between features, which is recorded as r. In this paper, pearson correlation coefficient is used to reflect the linear correlation degree of two features X and Y. The r value is between -1 and 1, and the greater the absolute value, the stronger the correlation.

We compute Pearson coefficients between any two features to filter out highly correlated features. Considering memory consumption and time performance, the best classification results are obtained by using the least features possible. In addition to the sample number and label, seven features are obtained, two features are obtained from the classification of attack models, so nine features are obtained.

5.3 Human Recognition Model Based on XGBoost

Random forest, gradient lifting trees, and convolution neural networks [9, 13, 19–21] have been used in human-machine identification and behavioral validation based on machine learning methods.

Table 1: Features and Descriptions

Feature	Description
x_max_target	Distance between maximum x-coordinate and target point
data_x_min	Minimum x-coordinate
delt_x_max	Maximum x-coordinate difference
acc_speed_x_start	X-coordinate initial acceleration
speed_x_end	X-coordinate end speed
data_y_start	Y-coordinate start value
speed_xy_start	Slide start speed
delt_xy_start	Slide start difference
delt_speed_t_start	The difference in the degree of change in starting time

Random forest models tend to get stuck in over-fitting in noisy sample sets, gradient-lifting tree serial training data takes time on larger datasets, and has lower classification performance than Extreme Gradient Boost (XGBoost). Convolutional neural networks are used to process graphical mouse sliding tracks and are not suitable for recognition in directional sliding scenes.

In this paper, XGBoost model algorithm is used as classifier. Compared with other machine learning algorithms, XGBoost not only has a good classification effect, but also can effectively prevent the occurrence of overfitting and improve the generalization ability of the model.

XGBoost is a variant of the GBDT algorithm. It is a commonly used supervised integrated learning algorithm with high flexibility, scalability and the ability to build parallel models. It supports a variety of weak learners such as gbtree, gblinear. XGBoost's tree model training is much better than linear models. Compared with the GBDT algorithm, XGBoost not only uses the first derivative information, but also uses the second-order Taylor expansion of the cost function. To prevent overfitting, regular terms are added to the target function of XGBoost. For example, formula (5):

$$res = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i) \tag{5}$$

XGBoost follows the forward iteration method, and the t th iteration of the sample i correlates with the predicted results of the (t-1) th tree. As in formula (6):

$$\hat{y}_i^{(t)} = \sum_{n=1}^{t} f_n(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{6}$$

XGBoost maintains high precision while performing parallel operation on the feature gain at the feature granularity, which significantly improves the fitting speed.Mouse sliding track may have different coordinate points at the same time point due to sampling. When calculating features, missing values will appear. Compared with other linear models, XGBoost model allows the existence of missing values. It can specify branch direction for missing data, which can solve the problem of missing data caused by sampling.

5.4 Evasion Attack Detection

Slide evasion attacks can be detected based on feature similarity between sliding tracks, but identical track characteristics are difficult to appear. A structure feature table is built to record the sample characteristics that have been predicted by the model, and the k nearest neighbor algorithm is used to calculate the feature similarity using Manhattan distance. Sliding tracks contain displacement points and time points. By calculating Manhattan distances for velocity characteristics and summing them up, tracks smaller than the threshold k are considered evasion attacks.

Threshold K is defined as the minimum difference of feature distances in the sample set. The speed-related features extracted in this paper include start, middle and end points. These speed features are not used as classification features, but they can be used for evasion detection by calculating the lateral and vertical speed characteristics of the training dataset. Starting speed, ending speed, average value and variance are selected as evasion attack detection features. In this paper, the Manhattan distance is calculated and summed for any two sample features in the training sample set, and the minimum value after removing the zero value is set to the threshold K.

In this paper, we propose an EDMD (Evasion Detection using Manhattan Distance) algorithm. If we traverse the structure feature table and calculate the difference value of each sample first, then the absolute value, and finally the sum, it will bring high time consumption. If vector operation is used, the efficiency can be greatly improved. When the sample vectors are subtracted from the structure characteristic table matrix in the same dimension, the feature difference of the sample records in the table can be obtained. The sum is made after the absolute value of the feature difference is calculated. The time required is much less than that of the former method, and the single sample detection time is 1.6 seconds when the data set is 100000. Through further optimization, it was found that the absolute value operation is time-efficient, and the time efficiency of comparison operation is higher than that of sum operation. The speed of comparison operation in matrix is much faster than that of single-run sum operation. Therefore, if there is a difference greater than k value in single-run operation, it can skip to the next run without performing sum operation and repeat the process. The single sample detection time for this method dataset was 0.05 seconds at 100,000 samples.

6 EXPERIMENTS AND RESULTS

6.1 Experimental Datasets

In this paper, two datasets are used, the first one is from mouse track desensitization data collected by a human-machine validation product[1], and the second one is from Shen[16] for user authentication. The first dataset contains 103,000 data sets, of which we used 3,000 samples as training datasets and 100,000 samples as test datasets. Of these, 3000 training samples contain 2600 human and 400 robot tracks.

The second data set was published by Shen[16] about a total of 17,400 samples of 58 people. Dataset 2 serves as a validation dataset to validate model generalization capabilities. The data format is operation code (512 is sliding under Windows system, 513 is left-button pressed, 514 is left-button raised), x-coordinate and y-coordinate with the upper left corner of the screen as the origin and time stamp information after desensitization. We extracted data sliding in eight directions from the Shen[16] dataset and converted it into a training dataset format field, resulting in 165,238 human behavior samples.

Algorithm 1 Evasion Detect.

```
Input: threshold $k$; Evasion_Feature $feat$; Evasion_lib $r$; Suspect_sample $s$;
Output: bool $res$
1: $absValue$ = abs($r - s$);
2: $greaterValues$ = where($absValue > k$);
3: $GValue$ = unique($greaterValues$);
4: $allLib$ = range(len($r$));
5: $SusIdList$ = difference($allLib$, $GValue$);
6: if $SusIdList$ not empty then
7:     for $id$ in $SusIdList$ do
8:         if sum($GValue[i]$) < $k$ then
9:             $res = 0$
10:        else
11:            $res = 1$
12:        end if
13:    end for
14: else
15:    $res = 1$
16: end if
17: return $res$;
```

We will process the second data set as the first data set format by: (1) Start at the press position of the left key and end at the lift position of the left key. The target point is the pixel difference relative to the starting point and the the timestamp is a recorded timestamp. (2) Tracks of each segment in each of the eight directions are extracted, and only X-axis forward tracks are extracted.

The training samples are collected in the same environment as the human data in the test samples, where the machine behavior attack pattern in the training dataset contains the attack pattern in the test dataset. The sliding path consists of a series of (x, y, t) three-dimensional coordinate points representing the location of the mouse at point in time (t) (x, y).The data is desensitized, time (t) does not correspond to the real time, and samples are taken at non-fixed intervals during mouse movement. The target coordinate is the target end point of the mouse movement. Because the real moving end point and target coordinate will be different, the end point of the track is not necessarily equal to the target coordinate. The focus is on the sliding x coordinate.

Table 2: Dataset Fields

Field	Type	Explanation
id	unsigned int	Sample number
trajectory	string	(x, y, t) path points separated by semicolons
target	string	Target location (x, y)
label	bool	1: Human trajectory;0: Machine trajectory

6.2 Data Preprocessing

There is a problem of unbalanced data proportion between training data set and test data set. We use the random sampling method to expand the sample set. As the mouse sliding behavior is a continuous trajectory, we randomly select 70% of the coordinates of each training sample as a new training sample and add it to the training data set. This process is repeated twice, and a total of 9000 training data sets are obtained, including 7800 human data and 1200 machine data. Jorgensen[8] put forward the problem of different attributes of normalized control samples in the literature. The reason why this paper did not normalize is that the training set and the test set are generated in the same environment. The trajectory path data has important distinguishing features in human-computer verification, and the data set contains the coordinates of the target points. Normalization will lead to sample distortion.

6.3 Experiment and Result

The program ran on a Linux server with a CPU of Intel core i7-8700k and memory size of 48G.

We use the XGBoost Library of Python language as a classifier. After extracting the sliding track features from the original dataset, we fill -100 with NaN value data to reduce the impact on the classification model. We used the Precision, Recall as the evaluation criteria for the experimental results, that is:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

TP refers to normal data that is correctly classified, FP refers to negative data that is incorrectly labeled as normal data, and FN refers to positive data that is incorrectly labeled as negative data.

We extract features from the Shen[16] dataset, because the dataset contains only positive samples, using the false rejection rate (FRR) as the evaluation indicator on the validation set, and the FRR calculation formula is (9), where N_{FA} is the number of false acceptances, N_{GRA} is the number of tests.

$$FRR = \frac{N_{FA}}{N_{GRA}} \tag{9}$$

6.3.1 *Feature Importance Analysis.*

With the optimal parameters conditions of the XGBoost model, in order to evaluate the impact of different features to the models, we use the evaluation model importance function in XGBoost to score features based on the influence of different features on the split of decision trees in the model. We select the top 20 features with the highest feature importance score and remove 13 features with high Pearson correlation coefficient and add 2 attacking model features.

Figure 10 shows the importance score of the remaining features and the attacking model features after removing the highly correlated features, which shows that the features have a higher importance score after removing the redundant features. When the slider verification code is sliding horizontally, the feature in the x-axis direction has a strong influence on the model, but the distinction between the time difference and the feature in the y-axis direction cannot be ignored.

We also compared the effects of eliminating redundant features before and after the classification, as shown in Table 3. Under the premise of using the attacking model features, there are 22 features before the correlation is removed. The high correlation features have a negative impact on the experimental results. It has the best classification effect after the redundancy is removed.

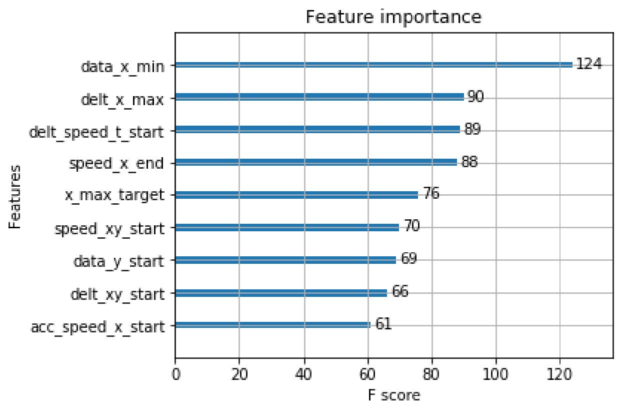

Figure 10: Feature importance score.

Table 3: Feature correlation experiments

Method	Precision	Recall
Before removing correlation	95.41	99.91
After removing correlation	98.94	99.86

6.3.2 Parameter Optimization.
By optimizing parameters of XGBoost, the model can achieve better classification results. When the booster is gbtree, the learning rate (eta) is set to 0.3, and the minimum number of leaves (min_child_weight) is adjusted to 1.5, the classification precision is depend on gamma value and maximum depth (max_depth).

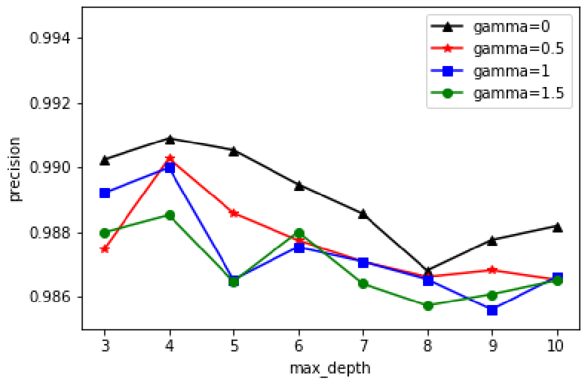

Figure 11: Effect of parameters gamma and max_depth on accuracy.

When gamma is set to 0, the decision tree will split as long as the loss function is reduced. Max_depth controls the lowest proportion of the total number of samples in a subtree, and if the tree is too deep, it will result in over-fitting. As Figure 11 shows, when the gamma value is 0, when max_depth is 4, the model is with higher precision. Table 4 shows parameter optimization increases the precision, recall.

Table 4: Parameter optimization

Method	Precision	Recall
Before adjusting parameters	98.94	99.86
After adjusting parameters	99.09	99.88

6.3.3 Dataset experiment.
This experiment uses random forest, support vector machines (SVM) and GBDT as comparison models, fits training data and predicts test datasets in the same experimental environment, each model parameter uses the default parameters of function library. Table 5 shows the results of the experiment. Experiments show that XGBoost is slightly better than random forest and GBDT and significantly higher than SVM in models that use current feature training with 100,000 datasets when the model uses default parameters.

Table 5: Algorithm evaluation on test dataset

Classification Model	Precision	Recall	Average time
GBDT	98.31	99.84	1.32s
SVM	96.96	98.95	1.10s
Random Forest	98.69	99.88	5.40s
XGBoost	98.94	99.86	0.23s

Shen[16] dataset contains eight tracks with sliding directions. Table 6 is the *FRR* on the validation set. The XGBoost method has the lowest error rejection rate. Shen[16] dataset contains eight tracks with sliding directions. In this paper, the feature is extracted as a lateral sliding feature, and more attention is paid to the importance of the feature in the X-axis direction. Human identification of sliding tracks in other directions may result in discrepancies. Table 6 also shows that when only 20678 X-axis tracks are extracted, the model has better human-machine identification.

Table 6: Algorithm evaluation on Shen[16] dataset

Classification Model	FRR	FRR(X-Axis forward only)
GBDT	13.28	1.09
SVM	100	100
Random Forest	13.27	0.93
XGBoost	13.26	0.93

6.3.4 Evasion Detection Experiment.
Due to data collection problems, deviations may occur in the entire track. We randomly selected 300 samples from the training dataset and compared the deviations between unbiased and different pixels, respectively. In a trajectory, only the horizontal sliding coordinate

points are offset. The result is shown in the figure12. The abscissa offset rate in the figure12 indicates that the coordinate points of X% are offset by 1 pixel or 3 pixels. On the premise of having the same K value, when having the same offset rate, the more offset pixels, the worse the evasion detection effect. To a complete replay behavior, i.e. 0% deviation rate, the replay behavior can be completely detected. Increasing the K value can effectively deal with the existing deviation, but it will bring greater time consumption. At the same time, due to the characteristics of the slider verification code, the deviation generally does not exceed 3 pixels, otherwise it will not coincide with the end point and is considered not a complete sliding path. However, it is difficult for machine learning methods to detect replay behavior, even if the tracks contain deviations. Therefore, a complete detection system needs to include machine learning detection model and evasion attack detection method.

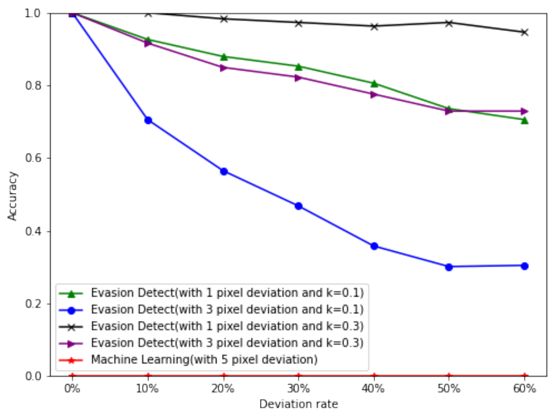

Figure 12: Evasion detection at different deviation rates.

We randomly selected 3000 horizontal sliding samples from Shen[16] data set and performed evasion detection, and 229 suspected evasion attacks were obtained. Our analysis of this phenomenon shows that the data set in Shen[16] is used for identity authentication, including multiple directional sliding of the same person and sliding of different people in the same coordinate position. When the selected fixed k value is large, it is prone to similar replay behavior.

6.4 Method Comparison

In addition to the above experimental content, we also compare other mouse behavior identification methods, table 7 includes other research methods using this human-machine authentication dataset and user authentication methods for mouse behavior.Kang[9] did not classify the machine behavior in detail, did not extract the specific attack model features from it, and there are interference features. Zhang's method[21] does not extract enough features. He considers the problem of feature correlation, but the classification results are not good because of too few features. Compared with other methods, we consider the correlation between features more and integrate the potential classification features of machine behavior.In the model selection, XGBoost model with faster speed and higher sensitivity of data set is used, which makes our method has better performance.

Table 7: Comparison of results

Article	Method	Year	Precision	Recall
OuYang[13]	GBDT	2017	91.03	91.06
Zhang[21]	GBDT	2018	94.27	-
Xu[19]	CNN	2019	92.20	-
Kang[9]	random forest	2021	97.83	94.72
Our Work	XGBoost	2021	99.06	99.88

Compared with identity authentication methods, human-machine identification has a higher classification accuracy. Shen[14, 16] has done several user authentication experiments using different methods, the FAR is 8.74, the FRR is 7.69, and the results of the methods become worse as the number of participants increases.In large-scale user authentication, mouse gliding behavior becomes less distinguishable.

From the fuzzy entropy, we can see that the uncertainty of human behavior is very high, so it is unavoidable to produce bias when sliding validation is used many times.When used for robot validation, the server-side privacy data can be better protected and unfair competition from malicious scripts can be avoided.Robot programs operate the mouse regularly and contain multiple repetitive tracks. In the main automation software, machine behavior has defects in imitating human behavior, so machine learning methods can better distinguish human behavior.

6.5 Time Efficiency Analysis

Under the current feature engineering, the model achieves good classification results. To test the time efficiency of identification methods, we fit the data set with model training to get the average completion time. Table 8 is the time spent in human-machine identification experiments on three different number of sample sets. The time efficiency of calculation features and evasion detection is analyzed separately. The evasion detection uses 274,238 samples after three data sets are merged. From the calculation efficiency, it shows that it takes less than 2 milliseconds to compute each feature. When using nine features for human-machine identification, the main time consumed in calculating the features is to complete one human-machine identification within 2ms, and to complete one evasion attack detection within 200ms, which meets the real-time requirements of commercial platforms. Compared with user authentication experiments[3, 6, 10, 12, 14–16, 18], human-machine identification is faster and more accurate, which brings great convenience to reduce user perception and improve the security of private data. About the evasion attack detection, as the number of sample sets increases, the time consumed for evasion attack detection increases linearly.

Table 8: Time efficiency

Sample set	Time spent	Time efficiency	Content
Training dataset 9000 samples	15.17s	1.69ms	Feature calculation
Test dataset 100,000 samples	143.50s	1.44ms	Feature calculation
Shen[16] Dataset 165238 Samples	222.75s	1.3ms	Feature calculation
Test dataset 100,000 samples	2894.15s	53ms	Evasion attack detection
Consolidated dataset 274238 samples	21382.58s	147ms	Evasion attack detection

7 DISCUSSION

Machine behavior of mouse sliding contains many attack modes. In this paper, human-mouse sliding behavior tracks are analyzed on existing datasets, and machine behavior is classified into five attack modes and their characteristics are analyzed. However, learning the characteristics of specific attack modes may have limitations. In order to improve the robustness of the model, it is necessary to learn more attack modes and extract strong classification features. Fitting the model and learning after dimension reduction of high-dimensional features has better classification results than learning from the features of specific attack modes.

Human-machine identification validation is time-sensitive, and using more advanced classification models will result in greater time consumption. Therefore, the extra time consumed is unacceptable if the increase in the correct rate is not significant.

evasion attack detection can lead to false positives in datasets over 100,000 levels, and the time consumed increases linearly. Regular refreshing of the comparison database can alleviate this situation.

8 CONCLUSION

This paper compares the behavior of sliding mouse controlled by machine program with that of real human behavior, finds different attack models, and then discusses the characteristics of each attack model. After extracting the features based on the physical quantity information of the sliding path, the features with good classification effect are selected by the model score, and the features with high Pearson correlation. When collecting the data of mouse sliding, the problem of missing data may appear. This paper uses the XGBoost model as a classifier because XGBoost specify the branch direction for missing values, which can solve the sample data missing problem well. We evaluate models on real datasets and validate model generalization capabilities on an open dataset. Experiments show that XGBoost can achieve 99.09% detection accuracy and 99.88% recall with fewer features in the classifier. Our study also found human and machine-mimicked human behavior samples in two similar tracks in the data, which suggests that the attacker used evasion attacks. In this paper, an evasion detection method is presented and the optimized method meets the requirements of high efficiency of evasion attack verification detection. Experiments show that the methods proposed in this paper can detect evasion attacks effectively.

Mouse sliding detection can be used as an additional detection method for intrusion detection. In practice, it can judge the security of access behavior together with other user identity authentication technology

ACKNOWLEDGMENTS

This research work is funded by Ant Financial through the Ant Financial Science Funds for Security Research.

REFERENCES

[1] 2017. Mouse Track Recognition. http://bdc.saikr.com/vse/bdc/2017/.

[2] Ismail Akrout, Amal Feriani, and Mohamed Akrout. 2019. Hacking google recaptcha v3 using reinforcement learning. *arXiv preprint arXiv:1903.01003* (2019).

[3] Margit Antal and Lehel Denes-Fazakas. 2019. User verification based on mouse dynamics: a comparison of public data sets. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 143–148.

[4] Margit Antal and Elöd Egyed-Zsigmond. 2019. Intrusion detection using mouse dynamics. *IET Biometrics* 8, 5 (2019), 285–294.

[5] Weiting Chen, Jun Zhuang, Wangxin Yu, and Zhizhong Wang. 2009. Measuring complexity using fuzzyen, apen, and sampen. *Medical engineering & physics* 31, 1 (2009), 61–68.

[6] Shen Fu, Dong Qin, Daji Qiao, and George T Amariucai. 2020. RUMBA-Mouse: Rapid User Mouse-Behavior Authentication Using a CNN-RNN Approach. In *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.

[7] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[8] Zach Jorgensen and Ting Yu. 2011. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 476–482.

[9] Lulu Kang, Xingrong Fan, Xizhu Wang, Xiaoya Yang, and Rui Ming. 2021. Mouse Track Recognition Based on Feature Group Hierarchy and Semi-supervised Learning. *Computer Engineering* 47, 04 (2021), 277–284.

[10] Masud Karim, Hasnain Heickal, and Md Hasanuzzaman. 2017. User authentication from mouse movement data using multiple classifiers. In *Proceedings of the 9th International Conference on Machine Learning and Computing*. 122–127.

[11] Lei Ma, Chungang Yan, Peihai Zhao, and Mimi Wang. 2016. A kind of mouse behavior authentication method on dynamic soft keyboard. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 000211–000216.

[12] G Muthumari, R Shenbagaraj, and M Blessa Binolin Pepsi. 2014. Mouse gesture based authentication using machine learning algorithm. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*. IEEE, 492–496.

[13] Zhiyou Ouyang and Xiaokui Sun. 2017. Behavioral Verification Code Human-Machine Recognition Based on Gradient Lifting Model. *NETINFO SECURITY* 9 (2017).

[14] Chao Shen, Zhongmin Cai, Xiaohong Guan, Youtian Du, and Roy A Maxion. 2012. User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security* 8, 1 (2012), 16–30.

[15] Chao Shen, Zhongmin Cai, Xiaohong Guan, Chao Fang, and Youtian Du. 2010. User Identity Authentication and Monitoring Based on Mouse Behavior Characteristics [J]. *Journal on Communication* 31, 7 (2010), 68–75.

[16] Chao Shen, Zhongmin Cai, Xiaohong Guan, and Roy Maxion. 2014. Performance evaluation of anomaly-detection algorithms for mouse dynamics. *computers & security* 45 (2014), 156–171.

[17] Jesús Solano, Christian Lopez, Esteban Rivera, Alejandra Castelblanco, Lizzy Tengana, and Martin Ochoa. 2020. SCRAP: Synthetically Composed Replay Attacks vs. Adversarial Machine Learning Attacks against Mouse-Based Biometric Authentication. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security* (Virtual Event, USA) *(AISec'20)*. Association for Computing Machinery, New York, NY, USA, 37–47. https://doi.org/10.1145/3411508.3421378

[18] Yi Xiang Marcus Tan, Alfonso Iacovazzi, Ivan Homoliak, Yuval Elovici, and Alexander Binder. 2019. Adversarial attacks on remote user authentication using behavioural mouse dynamics. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–10.

[19] Hongjun Xu, Hong Zhang, and Wei He. 2019. Cloud User Anomaly Detection Method Based on Mouse Behavior. *Journal of Harbin University of Science and Technology* 24, 04 (2019), 127–132.

[20] Zhen-yi Xu, Yu Kang, Yang Cao, and Yu-xiao Yang. 2019. Man-machine verification of mouse trajectory based on the random forest model. *Frontiers of Information Technology & Electronic Engineering* 20, 7 (2019), 925–929.

[21] Zhiteng Zhang and Linlan Liu. 2018. Mouse Trajectory Identification Method and Research Based on Gradient Promotion Decision Tree. *Information & Communications* 09 (2018), 17–19.