applied sciences

MDPI

Article

An Empirical Evaluation of Online Continuous Authentication and Anomaly Detection Using Mouse Clickstream Data Analysis

Sultan Almalki *, Nasser Assery and Kaushik Roy

Department of Computer Science, North Carolina A&T State University, Greensboro, NC 27411, USA; naassery@aggies.ncat.edu (N.A.); kroy@ncat.edu (K.R.)
* Correspondence: ssalmalki@aggies.ncat.edu

Abstract: While the password-based authentication used in social networks, e-mail, e-commerce, and online banking is vulnerable to hackings, biometric-based continuous authentication systems have been used successfully to handle the rise in unauthorized accesses. In this study, an empirical evaluation of online continuous authentication (CA) and anomaly detection (AD) based on mouse clickstream data analysis is presented. This research started by gathering a set of online mouse-dynamics information from 20 participants by using software developed for collecting mouse information, extracting approximately 87 features from the raw dataset. In contrast to previous work, the efficiency of CA and AD was studied using different machine learning (ML) and deep learning (DL) algorithms, namely, decision tree classifier (DT), *k*-nearest neighbor classifier (KNN), random forest classifier (RF), and convolutional neural network classifier (CNN). User identification was determined by using three scenarios: Scenario A, a single mouse movement action; Scenario B, a single point-and-click action; and Scenario C, a set of mouse movement and point-and-click actions. The results show that each classifier is capable of distinguishing between an authentic user and a fraudulent user with a comparatively high degree of accuracy.

Keywords: continuous authentication; anomaly detection; behavioral biometrics; mouse dynamics; machine learning; deep learning

check for updates

Citation: Almalki, S.; Assery, N.; Roy, K. An Empirical Evaluation of Online Continuous Authentication and Anomaly Detection Using Mouse Clickstream Data Analysis. *Appl. Sci.* **2021**, *11*, 6083. https://doi.org/10.3390/app11136083

Academic Editor: Juan Francisco De Paz Santana

Received: 8 April 2021
Accepted: 24 June 2021
Published: 30 June 2021

1. Introduction

In the current age of internet technology, authentication is a major security issue because authentication failures often cause detrimental effects. Recognition of masquerading is a crucial factor in preventing identity theft. Username/password-based applications routinely used in social networks, e-mail, e-commerce, and accessing online banking are not sufficiently protective technologies due to the increasing number of hacking attacks [1]. Theft of passwords can result in massive damage to individuals and companies. Computer security is becoming increasingly important for both individual users and society with the increase in communication between the services and devices in use. On online security systems, continuous authentication systems (CAs), and anomaly detection systems (ADs) are necessary to handle the rise in unauthorized access. Biometrics make the authentication stronger and uniquely determine a user's identity. The working principle of a biometric authentication system is to use a person's physical and behavioral traits. The physical biometrics used to detect fraudulent identification are the iris, the face, the fingerprints, and the voice. Their use is an intrusive method that provides sufficient authentication [1,2]. Physical biometrics are often considered accurate for authentication, but their use faces potential privacy concerns [3]. Behavioral biometrics on the other hand is an alternative approach to physical biometrics that provides an additional protection system against fraudulent identification [4,5]. The principle of behavioral biometrics is based on behavioral traits such as a person's handwriting patterns, gait, mouse dynamics, and keystrokes [6,7]. Behavioral biometrics is a quantifiable method that generates behavioral profiles of users.

Many behavioral biometrics studies have been done using CA and AD to detect intruders. Behavioral biometrics provide several advantages over traditional methods of authentication. For instance, a person's mouse behavior is not intrusive but is capable of providing continuous authentication. Furthermore, behavioral biometrics analysis of a person's keyboard usage behavior does not require access to the person's sensitive data. These advantages have led to an increase on research on the use of mouse dynamics biometrics in user-authentication systems [3,8,9]. Mouse dynamics is one of the behavioral biometrics that can save and analyze actions from a mouse input device such as general movement, drag-and-drop, and point-click actions while a person interacts with a specific graphical user interface [5,10].

Both online CA and AD techniques are other promising techniques that are capable of addressing the vulnerability in a static one-time authentication. These techniques cannot be used for a primary login, but they can be used to provide additional security, detect malicious actions, and mitigate the expected risk associated with the attacks and abnormal activities of online networks [11].

There are a lot of previous research experiments that have drawn conclusions on the basis of data only collected from a client's desktop, while there are a significantly lower number of experiments that focus on the online domain [12]. This study is restricted to the online domain for data collection by using an online game instead of other general GUI environments. A software program was developed to capture users' interaction via a pointing device based on two phases: a continuous authentication phase, and an anomaly detection phase. An empirical biometric-based study of different machine learning (ML) and deep learning (DL) techniques including decision tree classifier (DT), *k*-nearest neighbor classifier (KNN), random forest classifier (RF), and convolutional neural network classifier (CNN) was conducted in this research. The following are the contributions that this study provides:

- An overview of existing techniques related to CA and AD is given, as well as the methods in which these have been used.
- A new online mouse dynamics dataset was developed. Our dataset of 20 participants contained a combination of mouse movement and point-click actions.
- Approximately 87 features were extracted from raw mouse data.
- A new DL model for CA and AD that verifies the legitimacy of a user was developed.
- The results of extensive experiments conducted to validate different proposed approaches. Techniques of classifications including KNN, DT, RF, and CNN models were used. The proposed DL model achieved a high level of accuracy.
- A comparison of our work with the existing methods is given.

The remainder of this paper is organized as follows. Section 2 describes some related research for CA and AD. Section 3 provides a description of the data collection used in this research and the features extracted. Section 4 describes our approaches and classification techniques. Section 5 provides an implementation and experimental results. Section 6 presents the experimental evaluations. Section 7 has a summary of the discussion and conclusions.

2. Background and Related Work

As discussed in the introduction, an online network is subject to several types of security threats that can lead to intrusions. Moreover, irregular behavior on the Internet can indicate a security threat or other illegal behavior [13]. Biometric technologies can increase system security by measuring and analyzing the biological data of human behavior, then extracting hidden details from the acquired data and comparing the details to information stored in a recorded catalog. Biometric systems have been used to address security in a broad range of applications in several fields of our society, such as law enforcement, financial and trade security, information systems security, border control, and healthcare [14,15]. This section will briefly present some examples of research on the use of biometrics for continuous authentication and anomaly detection.

Chudá and Krátky (2014) [9] investigated user identification through online shopping activities. They first collected an extensive array of diverse data from 28 participants. They conducted their experiment using the *k*-nearest neighbor classifier. They achieved an accuracy of 87.5% by using 20 features extracted from mouse data. Authors in this study applied the traditional ML for classification using some handcrafted features. However, in our study, we compared the performance of the DL model with different ML algorithms.

The study of the behavioral components of human–mouse movement is generally referred to as mouse dynamics. Chuda et al. (2015) [16] studied user authentication in web users focusing only on point-click action. A comparison of cumulative distribution functions was used in their study. From 20 participants, the research extracted three mouse-click features: "pause to click", "click duration", and "pause after the click". They obtained 44% user authentication accuracy (ACC) when using features extracted from 100 mouse movement strokes, and they obtained 96% ACC when using features extracted from 100 mouse clicks.

Hamid et al. (2011) [17] developed a system to capture a user's characteristics using a set of random buttons. The system was built to capture the *x* and *y* coordinates of the mouse location and the time (in ms) of the event. Users must follow these buttons by moving the mouse and clicking on the buttons wherever the buttons appeared. A user must click 20 buttons to complete the experiment. After the user clicks the first button, a second button appears immediately in a random location. This step needs a user to do it six times to establish the user's personal profile under a controlled environment. The researchers' approach had data of five users with the same laptop and same mouse. They collected 30 files for all users, six files for each user. The researchers used a Euclidean distance algorithm for identification purposes. The experiment achieved 14 files that matched out of 30 files (46.67%). They attempted identification through hand/arm movements using mice and achieved an FRR of 53.55%. While a biometric system is developed for identification or verification as a part of authentication, the proposed approach in Hamid et al. (2011) focuses only on identification.

Hashia et al. (2005) [18] worked on mouse movement as a biometric. They proposed two authentication methods: the first method is for initial login of users (enrolment), and the second method is to monitor a computer for suspicious activities (verification). It requires from the user about 20 s to complete each of two methods. For the enrolment phase, a user must be using the mouse and following a series of dots that show one at a time on the user's screen. The purpose of this step is to record the coordinates of the mouse every 50 milliseconds and then calculate the speed, deviation from a straight line, and angles. They used the data collected from the enrolment phase for the verification phase by comparing a user's credentials and the data collected in the enrolment phase. They tested their approach using 15 participants of age 22–30. They achieved an error rate of 20% when using 1.5 standard deviations of the average from the corresponding enrolment value, and an error rate of 15 percent using 1 standard deviation of the average from the corresponding enrolment value. During the enrollment phase for passive authentication, they ran the program in the background to record the mouse coordinates for a shorter period of time; it was only 15 min for each participant.

Gamboa and Fred (2004) [19] developed a data acquisition system for collecting users' mouse activities. The system records all user interaction throughout the world wide web. The dataset was collected from 50 participants; each user had 400 strokes. A stroke is defined as a group of points between two actions. The authors proposed 58 behavioral features extracted from the raw data using some mathematical operations. These features were used to identify a user on the basis of how they interact with the system. Furthermore, the authors developed a sequential classifier using statistical pattern recognition techniques in order to distinguish between users. The authors achieved an equal error rate of 0.7% per 100 mouse strokes. The system used only a user's interaction characteristics, not the user's performance.

Pusara and Brodley (2004) [20] proposed a re-authentication approach using a user's mouse activities. They collected raw data from 11 student volunteers who spent about two hours on their own personal computers under an uncontrolled environment. The volunteers used Internet Explorer on a Windows computer for data collection. The experiment focused on using only Internet Explorer applications in order to decrease the difficulty of discriminating among the users' behaviors. The researchers used a supervised learning method, C5.0 decision trees. However, they split the dataset into 70% for training and 30% for testing only. They obtained an average false acceptance rate (FAR) of 1.75% and average false rejection rate (FRR) of 0.43%. The authors did not collect the client area mouse movements because their rate of occurrence was high.

Ahmed Awad E. Ahmed and Traore (2007) [1] divided the types of mouse actions into three categories: mouse movement (MM), point-click (PC), and drag-and-drop (DD). More recently, Ahmed and Traore (Ahmed & Traore, 2007, 2011) published a subsequent study based on the same dataset and using all three types of mouse actions. They collected their data from 22 participants over 998 sessions and conducted experiments on user authentication. Ahmed and Traore (2007) proposed a new form of behavioral biometrics via computer mouse dynamics: a detection technique using a neural network. They achieved relatively high false acceptance rate (FAR) of 2.4649% and a false rejection rate (FRR) of 2.4614%.

Ahmed and Traore (2011) [21] described the measurement results on an extended dataset of 48 users. They proposed a mouse dynamics biometric recognition system for commercial user identification. A fuzzy classifier technique was used to merge corresponding biometric scores. They reported results of a false acceptance rate of 0% and a false rejection rate of 0.36%. In contrast, some studies used only two types of mouse actions. Zheng et al. (2011) designed a reliable and effective continuous authentication mechanism by using only point-click (PC) and mouse movement (MM) actions, as defined by Ahmed and Traore (Ahmed Awad E. Ahmed & Traore, 2007). They used the support-vector machine (SVM) classifier for user verification. The results showed that their novel system to verify a user achieved an equal error rate (EER) of 1.3% using mouse clicks and 1.9% using mouse movements. Such performance does not meet the European standard for access control, which requires a commercial biometric system to achieve a FAR of less than 0.001% and a FRR of less than 1%.

Antal and Fejér (2020) [22] proposed a new one-dimensional convolutional network architecture using two datasets: the Balabit public dataset for performance evaluation [23], and the DFL dataset for transfer learning. (The DFL dataset was used to initialize the weights of our models). To avoid overfitting, they used the sigmoid activation function and a dropout layer with 0.15 probability. Moreover, their 1D-CNN model was trained in Keras using the Adam optimizer (learning rate: 0.002, decay: 0.0001, loss function: binary cross-entropy). They segmented the mouse dynamics data into fixed-size blocks and made two types of measurements: measurements using 300 blocks from each user (class-balanced measurement), and measurements using all blocks of data from each user (class-imbalanced measurement). They evaluated the model using three scenarios: (i) PLAIN models, trained from scratch using the training data from the Balabit dataset; (ii) TRANSFER1 models that use transfer learning, where the models were pre-trained on the DFL dataset; and (iii) TRANSFER2 models that were initialized with transfer learning, and then the weights were updated using the training data from the Balabit dataset. These were the results for number of blocks (300): PLAIN = 0.63, TRANSFER1 = 0.50, and TRANSFER2 = 0.66. These were the results for number of blocks (all): PLAIN = 0.55, TRANSFER1 = 0.34, and TRANSFER2 = 0.62. The Balabit public dataset was used in this study; it contains the mouse data from only 10 users, which may not be sufficient to produce a trustworthy and secure user model based on mouse dynamics.

Tan et al. (2019) [24] proposed different strategies that a potential attacker could use to carry out synthetically generated adversarial samples by using approaches based on imitation, a surrogate, or statistics. On the basis of the results of their experiments, they concluded that attacks based on neural networks perform better than statistics-based attacks. They discussed how the generation of mouse sequences is a difficult task to handle, and consequently the authors proposed that adversarial attacks have their flaws when carried out. The authors also elaborated on ways in which the robustness of these authentication models can be adversely affected, even when tested in a realistic way. In the latter half of the article, the authors showed the mechanism for accruing results from different experiments discussed in this paper. At the end of this article, they provide an overview of the extension of their surrogate-based attack approach.

Da Silva and Da Costa-Abreu [25] proposed a system of empirical biometric-based study for user identification using different neural networks in the online game League of Legends. The results of their experiments showed how different neural networks behave with the League of Legends biometric data and databases. The RBF and Bayesian networks indicated that it is possible to improve results by collecting the samples more often, despite the cost of processing. At the end, the authors state that future work can use a route where the separations between early-game, mid-game, and late-game samples are analyzed; this approach examines the user profile at each level as the same player carries out the game from beginning to end. Summing up this article, the authors mentioned that the traditional way of obtaining user verification by simply using email is not user-friendly because many AI-based features can be compromised if that approach is used. The amount of data that was collected was not sufficient to allow experiments to verify if a user is themselves attacking the account-sharing problem more precisely.

Table 1 provides a concise summary of the most important 10 studies on user authentication using mouse dynamics. The first column in each entry provides the source of the study. The remaining columns provide more information as given below:

- **Number of users:** The number of users that participated.
- **Data period:** The time of gathering the user's information.
- **Environment:** The place of collecting mouse behavior data.
- **Mouse action:** Characteristics of the actions received from the mouse input device for a specific user while interacting with a specific graphical user interface.
- **Type of Study:** Continuous authentication, intrusion detection, or static authentication.
- **Data used:** The dataset that was used for this study.
- **FAR:** False acceptance rate.
- **FRR:** False rejection rate.
- **EER:** Equal error rate.
- **Note:** Additional information.

Table 1. Previous characteristics of the most important existing works.

Paper	# User	Data Period	Environment	Action	Type of Study	Dataset Used	FRR	FAR	EER	Note
[26]	30	1–2 weeks	Controlled	MM-PC	Continuous Authentication	Collected	0.86%	2.96%	1.3%	25 clicks
[27]	25	N/A	Uncontrolled	MM-PC-DD	Continuous Authentication	Collected	17.66%	Not given	8.53%	30 actions
[22]	10	N/A	Uncontrolled	MM-PC-DD	Intrusion detection	Balabit	N/A	N/A	0.04%	20 actions
[28]	39	N/A	Controlled	MM	Static authentication	Collected	5.26%	4.59%	N/A	4 gestures
[29]	58	N/A	Controlled	MM-PC	Anomaly detection	Collected	N/A	N/A	11.63%	16.1 s
[30]	30	37 min	Uncontrolled	MM-PC	Continuous authentication	Collected	1.3%	1.3%	1.3%	20 clicks
[31]	28	5/10 min	Uncontrolled	MM-PC-DD	Continuous authentication	Collected	7.78–2.75%	9.45–3.39%	N/A	N/A
[32]	31	1 day	Controlled	MM-PC-DD	Continuous	Collected	2.10%	2.24%	N/A	N/A
[33]	52	One week	Uncontrolled	MM-PC-DD	Continuous	Collected	N/A	N/A	N/A	N/A
[20]	50	10–15 min	Controlled	MM-PC	Static authentication	Collected	2%	2%	0.020	50 strokes

3. Data Collection: The Cyber Identity and Biometrics Lab: Mouse Dynamics Dataset

For data collection, an approval for this research was received from the Institutional Review Board of the North Carolina A&T State University in order to ensure the suitability and ethicality use of human input data.

3.1. Dataset Mouse Recording Software

The software in this paper was developed using the Python language to gather various parameters of each individual user's mouse activities in a controlled environment. For this purpose, we used the pyHook package to provide callbacks for low-level global mouse and keyboard events using the Windows Hooking API [34]. The Hook Manager object was created in order to record mouse events. Once created, this Hook Manager was assigned callbacks for the collection of various events. In this data collection, all mouse events including Message Name, Message ID, Time, Window Name, X, and Y were recorded. The software runs as a background job; it starts monitoring mouse actions when the participants start performing the task. This software is able to obtain the events upon their occurrence and log these events in a log file (CSV format) that is continuously updated. This data collection software does not record any personal information of the student who is using the mouse. For data collection, the mouse-recording software was installed on one GPU computer and one identical computer from the Cyber Identity and Biometrics (CIB) lab at North Carolina A&T State University with a different kind of mouse device.

3.2. Participants

The CIB lab at the university was used for data collection. Due to a high degree privacy concern, the domain of volunteers was restricted to only the 20 participants who were invited personally; all the participants who participated in the data collection were from within the university. All participants were right-handed, with diversity in the categories of age (between 20 and 40), gender (male–female), educational background (bachelor's degree, master's degree, and Ph.D.), and nationality (e.g., United States, China, India, Nepal, and Saudi Arabia).

3.3. Running Participants

Data were collected by having participants play an online game called "Perfect Piano". The Perfect Piano game generates a set of random buttons and provides an opportunity for participants to move the mouse and click the buttons. The software was developed to capture data of a user when the user moved a mouse to follow a set of random buttons. Participants were informed the purpose of the task and given the training for completing the task. The team required participants to click rapidly and accurately by clicking the start button, then moving the mouse to click the buttons that are subsequently generated by the game. Each participant played the game individually for about half an hour, representing one data collection session for that individual user. For the process of data collection, the environmental settings (e.g., mouse pad, the position of the monitor, air conditioning, and position of the chair) were adjusted to each participant's preferred status in order to ensure the consistency of the purpose of the environmental parameter. In total, the team was able to collect a large dataset of approximately one million samples collected from 20 volunteers, consisting of 800,000 samples of mouse movement action and 200,000 samples of point-and-click action. Figure 1 shows the shape of the "Perfect Piano" game.

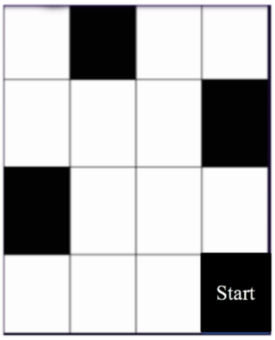


Figure 1. Perfect Piano game.

3.4. Raw Data Description

In this study, a controllable environment was established to gather the user behavior data. Regarding privacy, the data collection software did not record any personal information of the student who was using the mouse. For each session file, the team was able to collect a set of rows of data, where each row represents a recorded mouse action. Each recorded mouse event contains 7 parameters: Message Name, Message ID, Time, Window Name, X, and Y. The Message Name describes the event name (e.g., move left/right, click down/up). The Message ID represents the event ID (e.g., the event ID for mouse movement is 512, the event ID for mouse movement down button is 513, and the event ID for mouse movement up button is 514). The Time is the elapsed time in seconds since the start of the session being recorded. Window Name represents the name of the application being used (e.g., in our task, the event occurs in the Chrome web browser, and therefore the Window Name parameter shows the Chrome web browser). The X and Y parameters are the coordinates of the cursor on the screen.

3.5. Segmentation

A mouse movement action is a set of sequential user actions that represents a movement of the mouse between two screen locations. The screen object contains information about the event position. Figure 2 shows a mouse action, which consists of n events represented as a sequence of n points: {P1, P2, P3, ..., Pn} [22].

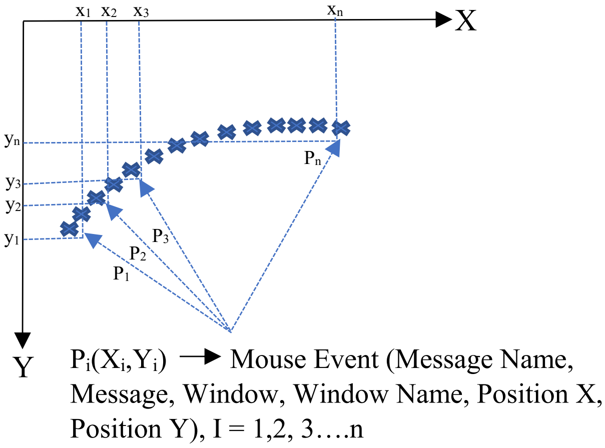
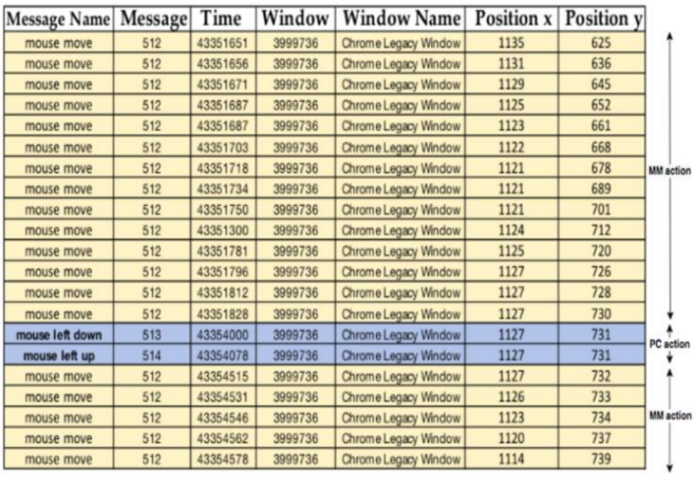

x1
x2
x3
xn
X
yn
y3
y2
y1
Pn
P3
P2
P1
Y
Pi(Xi,Yi) → Mouse Event (Message Name, Message, Window, Window Name, Position X, Position Y), I = 1,2, 3….n

Figure 2. Movement of the mouse between series of screen locations [22].

On the basis of the previous literature review, we divided the dataset of mouse actions into the three types of actions defined by Ahmed and Traore, namely, MM, PC, and DD [1]: MM describes a mouse movement between two screen locations; PC is a point-and-click, moving the mouse to a point and then clicking one of the mouse buttons; and DD is a drag-and-drop, a mouse drag movement initiated by pressing the primary mouse button and concluded by releasing it. In this study, the data of mouse actions were divided into two categories: (A) mouse movement, and (B) point-and-click action (press action and release action). The team consider an action as a point-and-click (PC) action when the previous event of a mouse is a mouse down and then mouse up; otherwise, it is a segment of data as a mouse movement (MM) action. Two complete segments of actions are shown in Figure 3.



Message Name	Message	Time	Window	Window Name	Position x	Position y
mouse move	512	43351651	3999736	Chrome Legacy Window	1135	625
mouse move	512	43351656	3999736	Chrome Legacy Window	1131	636
mouse move	512	43351671	3999736	Chrome Legacy Window	1129	645
mouse move	512	43351687	3999736	Chrome Legacy Window	1125	652
mouse move	512	43351687	3999736	Chrome Legacy Window	1123	661
mouse move	512	43351703	3999736	Chrome Legacy Window	1122	668
mouse move	512	43351718	3999736	Chrome Legacy Window	1121	678
mouse move	512	43351734	3999736	Chrome Legacy Window	1121	689
mouse move	512	43351750	3999736	Chrome Legacy Window	1121	701
mouse move	512	43351300	3999736	Chrome Legacy Window	1124	712
mouse move	512	43351781	3999736	Chrome Legacy Window	1125	720
mouse move	512	43351796	3999736	Chrome Legacy Window	1127	726
mouse move	512	43351812	3999736	Chrome Legacy Window	1127	728
mouse move	512	43351828	3999736	Chrome Legacy Window	1127	730
mouse left down	513	43354000	3999736	Chrome Legacy Window	1127	731
mouse left up	514	43354078	3999736	Chrome Legacy Window	1127	731
mouse move	512	43354515	3999736	Chrome Legacy Window	1127	732
mouse move	512	43354531	3999736	Chrome Legacy Window	1126	733
mouse move	512	43354546	3999736	Chrome Legacy Window	1123	734
mouse move	512	43354562	3999736	Chrome Legacy Window	1120	737
mouse move	512	43354578	3999736	Chrome Legacy Window	1114	739

Figure 3. Data collection parameters and mouse data segmentation into actions.

Moreover, Figures 4 and 5 show user behavior in terms of mouse movement and point-and-click actions belonging to User 1 and User 2, respectively.

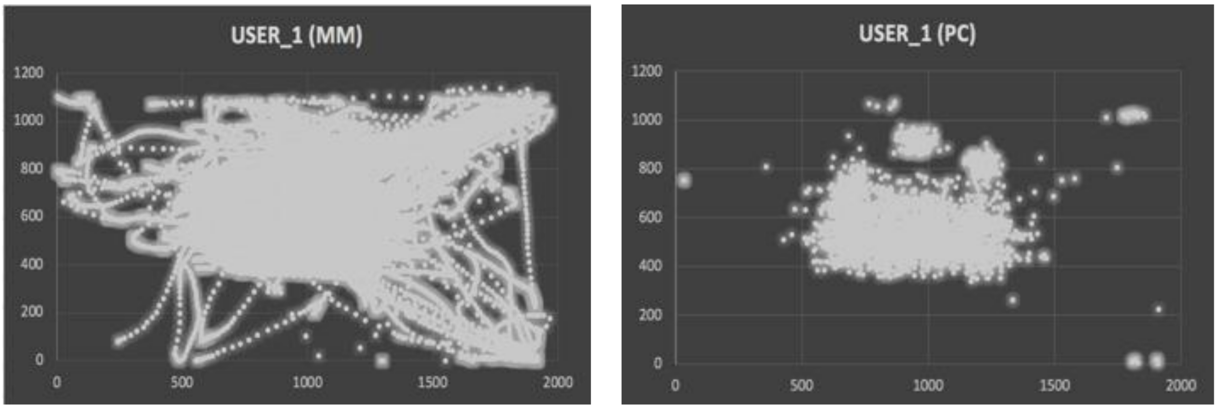


Figure 4. User 1 mouse movement, point-and-click actions.

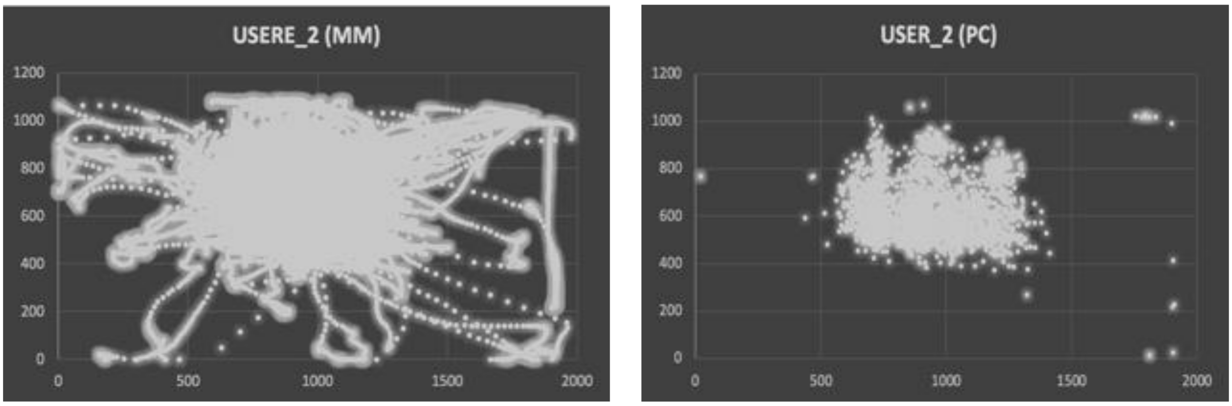


Figure 5. User 2 mouse movement, point-and-click actions.

3.6. Data Preprocessing

Feature extraction of mouse movement features and point-and-click features requires pre-processing and transformation of movement data into continuous time-series data. Continuous time-series data are generated from grouping the data by taking a Unix-timestamp column ("Time", in our case) as the grouping key so that a single Unix-timestamp can be linked to one or more records. At every Unix timestamp available in the dataset, the position at the x-axis is grouped and the mean is calculated. Similarly, the position at the y-axis is grouped with its respective timestamp, and the mean is calculated for the entire group of values belonging to a unique Unix timestamp. At the start of the processing of raw data, either mouse moment data or point click data are chosen. These data are grouped using timestamp as a key, and data aggregation is applied. These time-series-aggregated data are then resampled/up-sampled to make the dataset continuous. The sample time-series data are passed through a feature processing and extraction pipeline, which is specific to the type (mouse moment or point click) of data chosen in the first step. In the final step, we get hold of the required features that can be used for ML model training or data analysis. Figure 6 shows the steps of the data preprocessing.

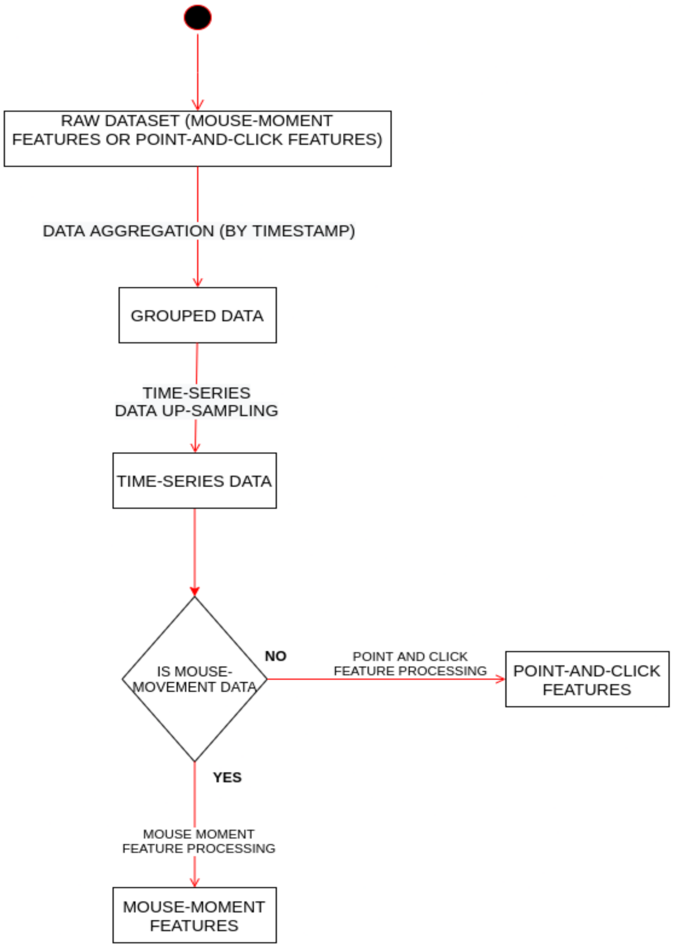

RAW DATASET (MOUSE-MOMENT FEATURES OR POINT-AND-CLICK FEATURES)
DATA AGGREGATION (BY TIMESTAMP)
GROUPED DATA
TIME-SERIES DATA UP-SAMPLING
TIME-SERIES DATA
IS MOUSE-MOVEMENT DATA
NO
POINT AND CLICK FEATURE PROCESSING
POINT-AND-CLICK FEATURES
YES
MOUSE MOMENT FEATURE PROCESSING
MOUSE-MOMENT FEATURES

Figure 6. Activity diagram of data preprocessing.

3.6.1. Time-Series Dataset Generation

Once the dataset contains a unique record for every Unix timestamp present in the dataset, the dataset is further applied through resampling over the time axis, where the time axis is the Unix timestamp. Resampling is done in such a way that all the missing records belonging to missing timestamps within the range of datasets are filled with interpolated values. This can be achieved by using functions of "pandas" in the Python library. Generally, resampling is followed by linear interpolation, which produces continuous time-series data that can be amicably used for complex time-series analysis. In this case scenario, resample and interpolation are used for completion of dataset. Data used for time series analysis need to have all data points either recorded or interpolated. The majority of time series techniques such AROMA/ARIMA and LSTM-based neural network training require data to be complete without any missing timestamps. Time series analysis on incomplete data with missing timestamps can lead to faulty or inaccurate analytical results.

3.6.2. Feature Extraction

In terms of empirical study, the output of the data acquisition models by itself is not sufficient to form a unique signature for each user. In this work, 87 features of individual mouse actions were extracted (15 from mouse movement actions, 72 from point-and-click actions) that can reflect individual behavior to a certain extent. This section briefly explains the features extracted from the raw data.

Mouse Movement Action Features

There are numerous movement-based features that can be harvested from a continuous time-series dataset: velocity of mouse pointer, acceleration, jerk, angle of movement, angular velocity, number of pixels travelled, curvature, and curvature change rate. Each feature is briefly described in the following sections. Velocity is the rate at which distance is changing per unit in a definite direction. Velocity along the x-axis, velocity along the y-axis, the magnitude of velocity over the mouse plane, and angular velocity were computed on the basis of *x(t)*, *y(t)* and Θ*(t)* series using the following Equations (1)–(4), respectively.

$$v(x,t) = \frac{x(t) - x(t-1)}{t - (t-1)} \tag{1}$$

$$v(y,t) = \frac{y(t) - y(t-1)}{t - (t-1)} \tag{2}$$

$$v(pixels,t) = \sqrt{v(x,t)^2 + v(y,t)^2} \tag{3}$$

$$W = (\Delta^{\rightarrow}\theta / \Delta^{\rightarrow}t) \tag{4}$$

In addition, acceleration is defined as change in velocity per unit time. Acceleration-based features are acceleration along the x-axis, acceleration along the y-axis, and magnitude of acceleration over the mouse plane, respectively, Equations (5)–(7).

$$a(x,t) = \frac{v(x,t) - v(x,t-1)}{t - (t-1)} \tag{5}$$

$$a(y,t) = \frac{v(y,t) - v(y,t-1)}{t - (t-1)} \tag{6}$$

$$a(x,y,t) = a(pixels,t) = \sqrt{a(x,t)^2 + a(y,t)^2} \tag{7}$$

Another used feature was jerk. Jerk is defined as a change in acceleration per unit time. The jerk along the x-axis, jerk along the y-axis, and magnitude of acceleration over the mouse plan was computed using the following Equations (8)–(10):

$$jerk(x,t) = \frac{a(x,t) - a(x,t-1)}{t - (t-1)} \tag{8}$$

$$jerk(y,t) = \frac{a(y,t) - a(y,t-1)}{t - (t-1)} \tag{9}$$

$$jerk(x,y,t) = \sqrt{jerk(x,t)^2 + jerk(y,t)^2} \tag{10}$$

Moreover, angular movement refers to direction of movement at a given timestamp. The **atan function(Θ)** using the x and y sequences was calculated. **Atan function(Θ)** is the angle of the path tangent with the *x*-axis. It can be calculated by the Equation (11) given below.

$$\theta = a\tan(\Delta^{\rightarrow}y/\Delta^{\rightarrow}x) \tag{11}$$

The **atan function** is a trigonometric function that is a common variation of the standard arctangent function; it can be used to define values related to a right triangle. Practically, the **atan** trigonometric function can be used to determine distances that are difficult to measure and produces results in the range (−π, π) (Figure 7) [22].

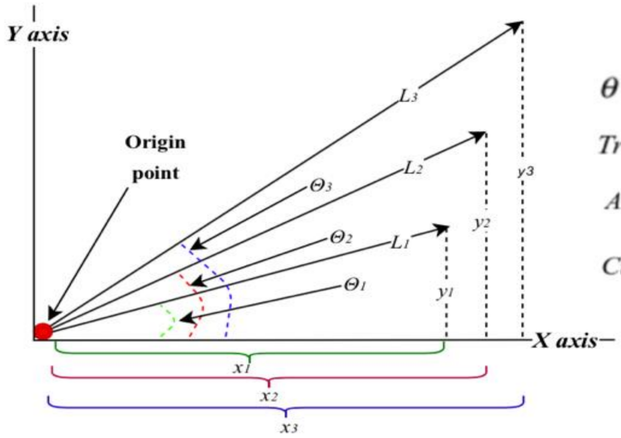

Y axis
Origin point
L3
L2
L1
Θ3
Θ2
Θ1
y3
y2
y1
X axis
x1
x2
x3
θ = a tan(Δ→y / Δ→x)
Travelled(pixels) = √((Δx)² + (Δy)²)
Angle of Curvature(c) = Δθ/ΔL
Curvature change rate(c) = ΔC/ΔL

Figure 7. Angular movement (Θ), travelled distance, angle of curvature and its rate of change.

In order to reduce the possible direction values, we used the 8 main directions defined by Ahmed and Traore (Figure 8) [1].

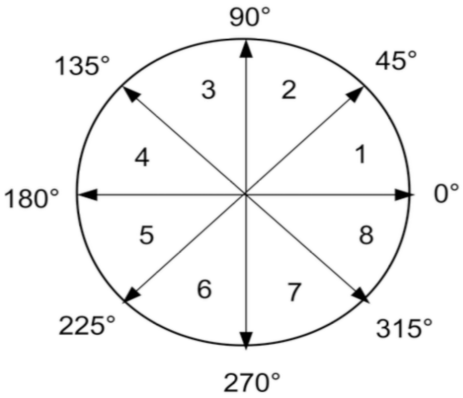


Figure 8. The eight directions. Angles between 0° and 45° fall into the direction 1.

The speed(s) also was calculated for each action as the ratio of the travelled. Distance travelled refers to the number of pixels travelled within a second (Figure 7). It can be calculated by Equation (12).

$$Travelled(pixels) = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{12}$$

Curvature distance is a ratio of distances in three consecutives recorded points on the screen. When a change in angular of the tangent per pixel divided by the length of the path from the origin point is found, the angle of curvature is computed (Figure 7). It can be calculated by the following Equation (13):

$$Angle\,of\,Curvature(c) = \frac{\Delta\theta}{\Delta L} \tag{13}$$

The curvature change rate was also calculated using Equation (14):

$$Curvature\ change\ rate(c) = \frac{\Delta C}{\Delta L} \tag{14}$$

Curvature change rate refers to the change in curvature per pixel travelled divided by the length of the path from the origin point (Figure 8).

Point-and-Click Action Features

Point-and-click features were extracted from mouse-based press actions and release actions. On the basis of press actions and release actions, we extracted two different datasets separately from the main dataset. The first extracted dataset contained only samples where the action is equal to "PRESS". The second dataset contained only samples where the action is equal to "RELEASE". Some of the features were taken from the main dataset as mouse movement features: time, velocity along the x-axis, velocity along the y-axis, magnitude of velocity over the mouse plane, acceleration over the mouse plane, jerk over the mouse plane, position-x (position of the pointer along the x-axis), position-y (position of the pointer along the y-axis), the angular velocity of the mouse pointer, and curvature. Given such features, a new combined dataset can be generated by merging the first dataset and second dataset along the columnar axis. Carefully looking at the samples in the newly generated dataset, we observed that each sample contained information related to the journey, starting from the press of the mouse button until the release of the mouse button. The newly generated dataset can be used to generate some interesting features related to each sample that involve total angular movement, absolute distance travelled, length of the trajectory (total distance travelled throughout the journey of the mouse pointer), straightness of trajectory, and elapsed time (total trajectory time). Total angular movement is defined as the total changes in the angles of the path during the journey between the press and release actions. The total angular movement was computed using the following Equation (15):

$$Total\ Angular\ Movement = \sum_{i=1}^{n} \theta i \tag{15}$$

where θ_i refers to the angular movement at the ith action in the trajectory.

In every single row, there are several datapoints between the press and release actions. On the basis of these datapoints, we then calculated the distance between the first and last datapoint positions, which is determined by Equation (16):

$$Absolute\_Dis\tan ce = \sqrt{(x1 - y1)^2 + (x2 - y2)^2} \tag{16}$$

where $x1$, $y1$ refers to the coordinates of the mouse pointer at press time, and $x2$, $y2$ refers to the coordinates of the mouse pointer at release time.

In addition, the length of the trajectory was computed between all sequence of points belonging to the trajectory. In Figure 9, the total trajectory length would be equal to sum of distance1 + distance2 + distance3 + distance4, where "distance" here refers to Euclidean distance. The length of the trajectory was computed using the following Equation (17):

$$Trajectory\ length = \sum_{i=1}^{n} \sqrt{(x1_i - y1_i)^2 + (x2_i - y2_i)^2} \tag{17}$$

where $(x1_i,\ y1_i)$ refers to the ith coordinate among n coordinates that belong to the trajectory for the given sample, and $(x2_i,\ y2_i)$ refers to $(i - 1)$th coordinate among n coordinates that belong to the trajectory for the given sample.

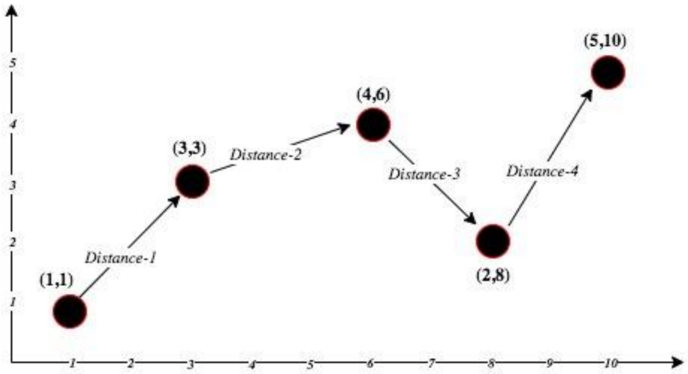


Figure 9. Length of trajectory.

Straightness of trajectory was computed as the ratio of the curve of trajectory during the journey between the press and release actions, which is inverse of curviness. The path between press and release was measured to see if it was completely straight; in that case, straightness would be 1, which is highest. Otherwise, if the path was not straight, the straightness of trajectory would be measured using the following Equation (18):

$$Straightness = \frac{Absolute\ distance}{length\ of\ trajectory} \tag{18}$$

The other extracted feature was elapsed time. Elapsed time is the time that has passed between the press action and the release action, computed using the following Equation (19):

$$Elapsed\ time = Release\ Time - Press\ Time \tag{19}$$

where "Release Time" refers to the time when mouse pointer was released, and "Press Time" refers to the time when mouse pointer was pressed.

Other features based on common statistics were extracted, such as min, max, mean, standard deviation, and variance for features such as velocity, acceleration, jerk, curvature, and angular velocity. Moreover, statistics such as min, max, mean, standard deviation, and variance can be calculated over an array of points belonging to the trajectory of every sample. Such an array of points can be extracted from the original dataset. This can be achieved by sorting the original dataset using time, followed by taking all samples between each given Press Time and Release Time. Now statistical operations can be performed over such an array of samples.

$$Mean = mean(x_{i\ldots n}) \tag{20}$$

$$Standard\ deviation = std(x_{i\ldots n}) \tag{21}$$

$$Minimum = \min(x_{i\ldots n}) \tag{22}$$

$$Maximum = \max(x_{i\ldots n}) \tag{23}$$

$$Variance = (standard\ deviation)^2 \tag{24}$$

where x can be any feature including velocity along the x-axis, velocity along the y-axis, velocity along mouse plane, acceleration over the mouse plane, jerk over the mouse plane, curvature, or angular velocity. In addition, the type of actions such as mouse movement action, press action, and release action were calculated as features. Table 2 shows all the features extracted from mouse movement and point-and-click actions.

Table 2. Features extracted from mouse movement and point-and-click actions.

Name	Mouse Movement Action	Point and Click Action		# Features
		Press Action	Release Action	
Velocity along x-axis	✓			1
Velocity along x-axis (mean, max, min, SD, variance)		✓	✓	10
Velocity along y-axis	✓			1
Velocity along y-axis (mean, max, min, SD, variance)		✓	✓	10
Velocity over the mouse (x-y) plane	✓			1
Velocity over the mouse (x-y) plane (mean, max, min, SD, variance)		✓	✓	10
Angular velocity	✓			1
Acceleration along x-axis	✓			1
Acceleration along y-axis	✓			1
Acceleration over the mouse (x-y) plane	✓			1
Acceleration over the mouse (x-y) plane (mean, max, min, SD, variance)		✓	✓	10
Jerk along x-axis	✓			1
Jerk along y-axis	✓			1
Jerk over the mouse (x-y) plane	✓			1
Jerk over the mouse (x-y) plane (mean, max, min, SD, variance)		✓	✓	10
Angular movement	✓	✓	✓	3
Distance travelled	✓	✓	✓	3
Angle of curvature	✓			1
Curvature change rate	✓			1
Curvature change rate (mean, max, min, SD, variance)		✓	✓	10
Length of trajectory		✓	✓	2
Straightness of trajectory		✓	✓	2
Elapsed time		✓	✓	2
Mouse movement action	✓			1
Press action		✓		1
Release action			✓	1
Total				87

4. Methodology and Behavioral Biometrics Model

We here describe the classification techniques used in this research. We applied separate classification techniques for different modalities. This research proposes a continuous authentication and anomaly detection system using mouse behavioral biometrics; this system can be applied to various online networking platforms. Each of four classification techniques is applied in order to distinguish a normal user from an abnormal user using the 87 features extracted from the raw mouse data. These are the machine learning and deep learning algorithms used to monitor the behavior of users: decision tree learning (DT), *k*-nearest neighbor (k-NN), random forest (RF), and convolutional neural network (CNN). In terms of the CNN model, our model differs from [35]; their CNN model has 2 convolutional layers and 3 fully connected layers. In contrast to [35], this model uses three types of layers to build the CNN architecture: convolutional, pooling, and fully connected (Figure 10).

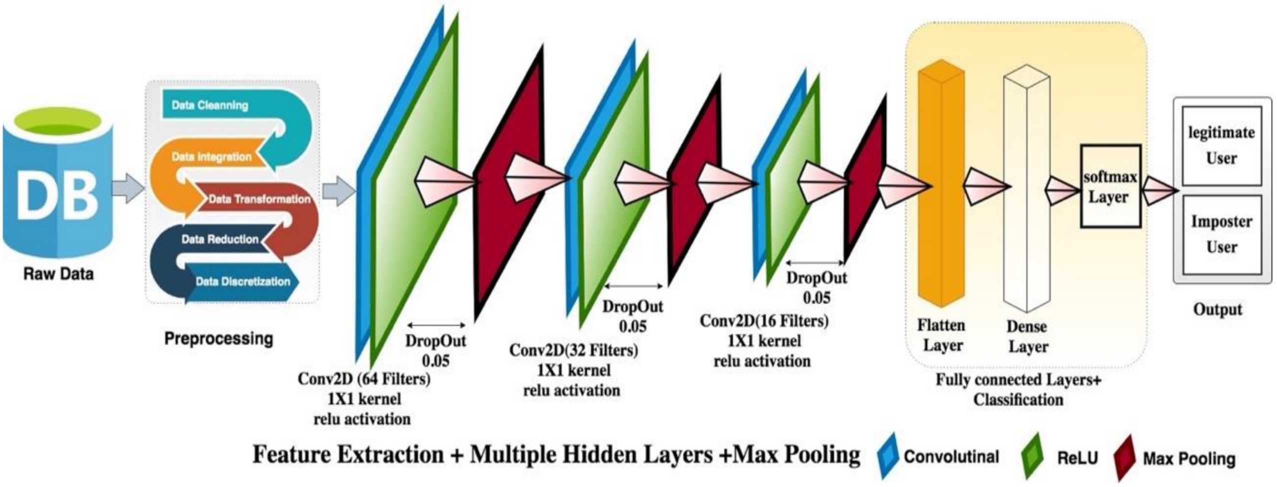


Figure 10. CNN architecture.

The first layer consists of 64 filters, followed by a second layer with 32 filters, and the last layer contains only 16 filters. A kernel of (1 × 1 pixels) is used for each of the three layers. Moreover, the activation function of the rectified linear unit (ReLU) is used. To reduce overfitting on the training set, we use a dropout probability of $p = 0.05$ between each two layers. All the three convolution layers and the max-pooling layer are connected with a fully connected (FC) layer to determine the final probabilities for each user. A pooling layer is located between every convolutional layer and fully connected layer pair. In Figure 11, the proposed user behavioral biometrics model consists of four modules: a data capture module, a feature extraction module, a classifier module, and a continuous authentication and anomaly detection module. The model is responsible for deciding whether some amount of mouse data belong to a given user. Specifically, the following steps describe how the proposed model works:

- Data collection phase: Raw data of the users are collected.
- Features extraction phase: Pandas and numpy were used for feature extraction.
- Data preparation phase: For the training phase, all the users' data were aggregated and put in random order. The training dataset was then split into two parts: the first part (80% of the data) was used for training, and the second part (20% of the data) was used for testing the model's performance. For every experiment, the balance of training sets and evaluation sets remained the same in order to avoid classifier bias.
- Select a classifier phase: DT, RF, KNN, and CNN were utilized to show the ability of the proposed model to determine whether a user was genuine or an impostor from a user's mouse clickstream data.
- Training data phase: The training process began by reading the characteristics of all the users from the training dataset and then loading them into the four classifiers to train the model. This step was a significant step since the training data contained the user behavior itself and a class label.
- Testing data phase: After completion of the training step, the model was tested on the new data that were never used for training, in order to categorize whether the user was a genuine user or an impostor.

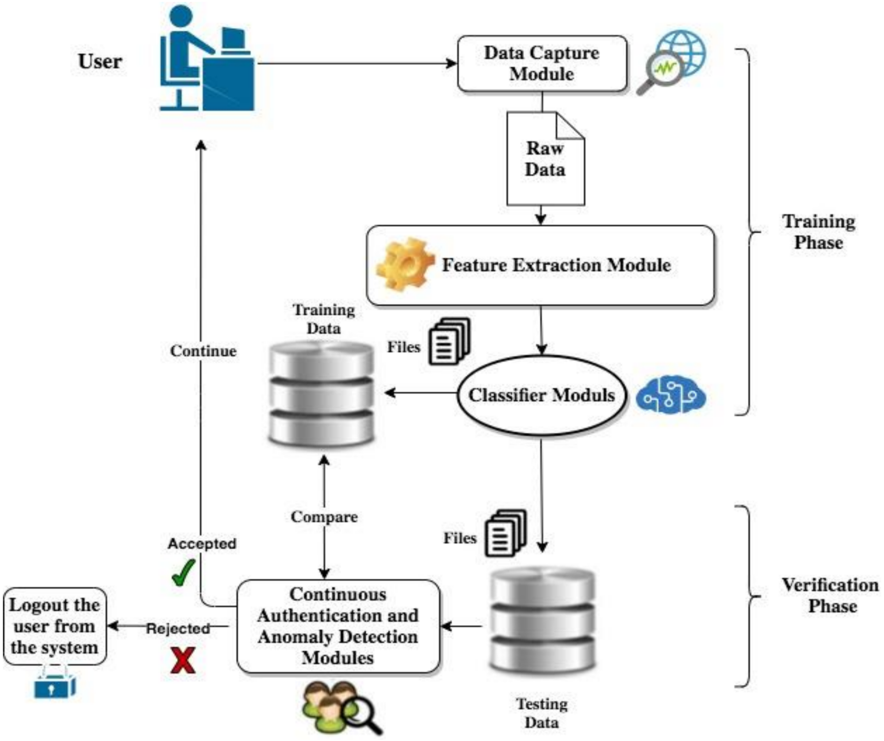


Figure 11. Continuous authentication and anomaly detection models.

5. Implementation and Experiment Results

This section presents the experiment setup and the analysis of results. The team conducted a set of experiments in order to validate the effectiveness of machine learning and deep learning techniques (decision tree (DT), *k*-nearest neighbor (KNN), random forest (RF), and convolutional neural network (CNN) classifiers) using the 87 features extracted from the raw mouse data. To avoid classifier biases, we trained all the experiments by using balanced training sets and evaluation sets. Scikit-learn software tools were used for the analysis of user behavior data [36]. The evaluations were measured using classifier accuracy (ACC), recall, precision, and F1-score. The experiments were conducted on the basis of two phases: (1) continuous authentication phase, and (2) anomaly detection phase. For each phase, each of the classifiers was trained and tested separately using each of these scenarios: a single mouse movement action (Scenario A), a single point-and-click action (Scenario B), and a set of mouse movement and point-and-click actions (Scenario C). We report evaluation results in terms of classification accuracy (ACC), area under the curve (AUC), false negative rates (FNRs), false positive rates (FPRs), and equal error rates (EERs). Besides these values, we report the performance of our detection system by receiver operating characteristic (ROC) curves.

5.1. Phase 1: Continuous Authentication Phase

The main idea of a continuous authentication system is to collect real-time user behavior information using mouse dynamics and use this information to continuously authenticate and reverify the user's identity. In particular, the behavior of the current user will be compared with the data stored in the system's database of the behavior of the genuine user. On the basis of the result of that comparison, the system either trusts the user and allows the user to continue working on the device, or the system logs the user out of the system and requires static authentication of the user in order for the user to continue working [37]. The dataset was separated into two parts: the first part (80% of the data) was used as a description of a genuine user in order to train the models; the second part (20% of the data) was used as actions of a user to be authenticated in order to test the model's performance. In this phase, experiments were conducted using each of the classifiers (*k*-

nearest neighbor (KNN), decision tree (DT), random forest (RF), and convolutional neural network (CNN)) for all users with a single mouse movement action, a single point-and-click action, and a set of mouse movement and point-and-click actions. Comparing the results obtained from all the experiments, we found that Scenario A (a single mouse movement action) achieved the highest accuracies compared to Scenarios B and C, with Scenario A KNN ACC: 98.0%, DT ACC: 94.6%, RF ACC: 97.9%, and CNN ACC: 98.8%. It was also found that the CNN model obtained the highest accuracy, recall, precision, and F1-score, with CNN ACC: 98.8%, recall: 96.3%, precision: 97.9%, and F1-score: 95.5%. Tables 3–5 illustrate all the results that were achieved.

Table 3. Phase 1: Continuous authentication results for Scenario A: single mouse movement action.

Classifier	KNN	DT	RF	CNN
Accuracy	98.0	94.6	97.9	98.8
Recall	93.8	93.7	94.2	96.3
Precision	95.9	92.8	96.0	97.9
F1-score	90.8	92.7	94.4	95.5

Table 4. Phase 1: Continuous authentication results for Scenario B: single point-and-click action.

Classifier	KNN	DT	RF	CNN
Accuracy	82.7	83.3	96.1	95.2
Recall	80.2	82.7	95.3	95.4
Precision	81.1	81.5	93.7	94.2
F1-score	82.5	82.6	93.9	95.1

Table 5. Phase 1: Continuous authentication results for Scenario C: set of mouse movement and point-and-click actions.

Classifier	KNN	DT	RF	CNN
Accuracy	96.7	94.2	97.2	96.9
Recall	91.2	90.2	97.1	94.4
Precision	93.1	93.7	93.3	92.2
F1-score	95.5	93.6	92.9	90.1

5.2. Phase 2: Anomaly Detection Phase

The basic idea of the anomaly detection system depends on the user's actions on the system. When the system compares the user's current behavior to the database of the user's behavior, the system should allow the user to keep working if there is only a small deviation in the user's current behavior; otherwise, the system must log out the user and require static authentication of the user before the user can continue working. In this phase, the dataset was split into 80% as genuine actions and 20% as impostor actions. The 20% of impostor actions were selected from the other users. To verify the feasibility of the proposed algorithms, we separately applied each of the four classifiers to three scenarios: a single mouse movement action (Scenario A), a single point-and-click action (Scenario B), and a set of mouse movement and point-and-click actions (Scenario C). The results showed that the highest accuracy was achieved also in Scenario A (a single mouse movement action) with KNN ACC: 98.2%, DT ACC: 92.2%, RF ACC: 98.0%, and CNN ACC: 98.5%. Individually, the CNN classifier in Scenario A obtained the highest accuracy, recall, precision, and F1-score, with ACC: 98.5%, recall: 97.3%, precision: 97.1%, and F1-score: 95.7%. Tables 6–8 show all the results.

Table 6. Phase 2: Anomaly detection results in Scenario A: single mouse movement action.

Classifier	KNN	DT	RF	CNN
Accuracy	98.2	92.2	98.0	98.5
Recall	97.8	90.7	96.2	97.3
Precision	93.9	90.8	97.0	97.1
F1-score	95.6	91.7	95.4	95.7

Table 7. Phase 2: Anomaly detection results in Scenario B: single point-and-click action.

Classifier	KNN	DT	RF	CNN
Accuracy	93.1	83.5	95.4	88.6
Recall	90.2	80.7	93.1	85.2
Precession	89.1	81.6	92.8	86.8
F1-score	91.5	81.4	93.6	82.9

Table 8. Phase 2: Anomaly detection results in Scenario C: Set of mouse movement and point-and-click actions.

Classifier	KNN	DT	RF	CNN
Accuracy	89.4	74.1	92.1	85.7
Recall	87.2	72.2	91.1	85.4
Precision	86.7	71.7	92.4	84.2
F1-score	84.3	72.6	90.2	83.1

6. Experiment Evaluation

This section evaluates the performance of each classifier on the basis of results achieved in the continuous authentication phase and in the anomaly detection phase. The performance of each method is measured using three common metrics: false acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER). FAR is the probability that a user who should be rejected is accepted by the system. FRR is the probability that a user who should be accepted is rejected by the system. EER is a threshold value between the false acceptance rate and the false rejection rate. Another important evaluation tool to examine the classifiers and show the performance of the biometric system is to plot the receiver operating characteristic (ROC) curves. It is important to interpret a classifier with the structure of its ROC curve because they show the prediction success of the models on several operating points corresponding to all possible thresholds. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) [38]. The following expressions were used for performance evaluation purposes [39]: TP: true positive, TN: true negative, FP: false positive, FN: false negative, FAR: false acceptance rate, FRR: false rejection rate, and EER: equal error rate:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

$$TPR = \frac{TP}{TP + FN} \tag{26}$$

$$TNR = \frac{TN}{TN + FP} \tag{27}$$

$$FPR = \frac{FP}{FP + TN} \tag{28}$$

$$FNR = \frac{FN}{FN + TP} \tag{29}$$

$$FAR = \frac{Number\ of\ accepted\ imposters}{Total\ number\ of\ imposters} \tag{30}$$

$$FRR = \frac{Number\ of \text{ rejected genuines}}{Total\ number\ of \text{ genuines}} \tag{31}$$

$$EER = \frac{FAR + FRR}{2} \tag{32}$$

6.1. Continuous Authentication Evaluation

The results of Scenario A (a single mouse movement action), Scenario B (a single point-and-click action, and Scenario C (a set of mouse movement and point-and-click actions) were evaluated using FAR, FRR, EER, and the ROC curve. The corresponding FAR, FRR, and EER were achieved. The following were the lowest FAR results: Scenario A (KNN: 0.009%), Scenario B (RF: 0.012%), and Scenario C (DT: 0.002%). The following were the lowest FRR results: Scenario A (KNN: 0.182%), Scenario B (RF: 0.027%), and Scenario C (DT: 0.007%). The following were the lowest EER results: Scenario A (CNN: 0.021%), Scenario B (CNN: 0.107%), and Scenario C (DT: 0.005%). The detailed results in terms of FAR, FRR, and EER are reported in Tables 9–11. In addition, ROC curves are plotted in Figures 12–14. We found that for CA, the DT achieved the lowest EER of 0.005% for MM and PC actions.

6.2. Anomaly Detection Evaluation

This evaluation is similar to the continuous authentication evaluation. The evaluation is presented for Scenarios A, B, and C using FAR, FRR, EER, and the ROC curve. We noted that Scenario A (single mouse movement action) had the lowest EER values: KNN EER: 0.045%, DT EER: 210%, RF EER: 0.035%, and CNN EER: 0.032%, compared with Scenarios B and C. The results for all scenarios are reported in Tables 12–14. The ROC curves are shown in Figures 15–17. The results show that for AD, the CNN achieved the lowest EER of 0.032% for single MM action.

Table 9. CA evaluation—Scenario A (single mouse movement action): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.009%	0.011%	0.010%	0.052%
FRR	0.182%	0.670%	0.208%	0.999%
EER	0.028%	0.327%	0.023%	0.021%

Table 10. CA evaluation—Scenario B: (a single point-and-click action): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.025%	0.015%	0.012%	0.051%
FRR	0.513%	0.311%	0.027%	0.918%
EER	0.229%	0.220%	0.068%	0.107%

Table 11. CA evaluation—Scenario C: (set of mouse movement and point-and-click actions): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.031%	0.002%	0.032%	0.052%
FRR	0.608%	0.007%	0.634%	0.930%
EER	0.222%	0.005%	0.155%	0.094%

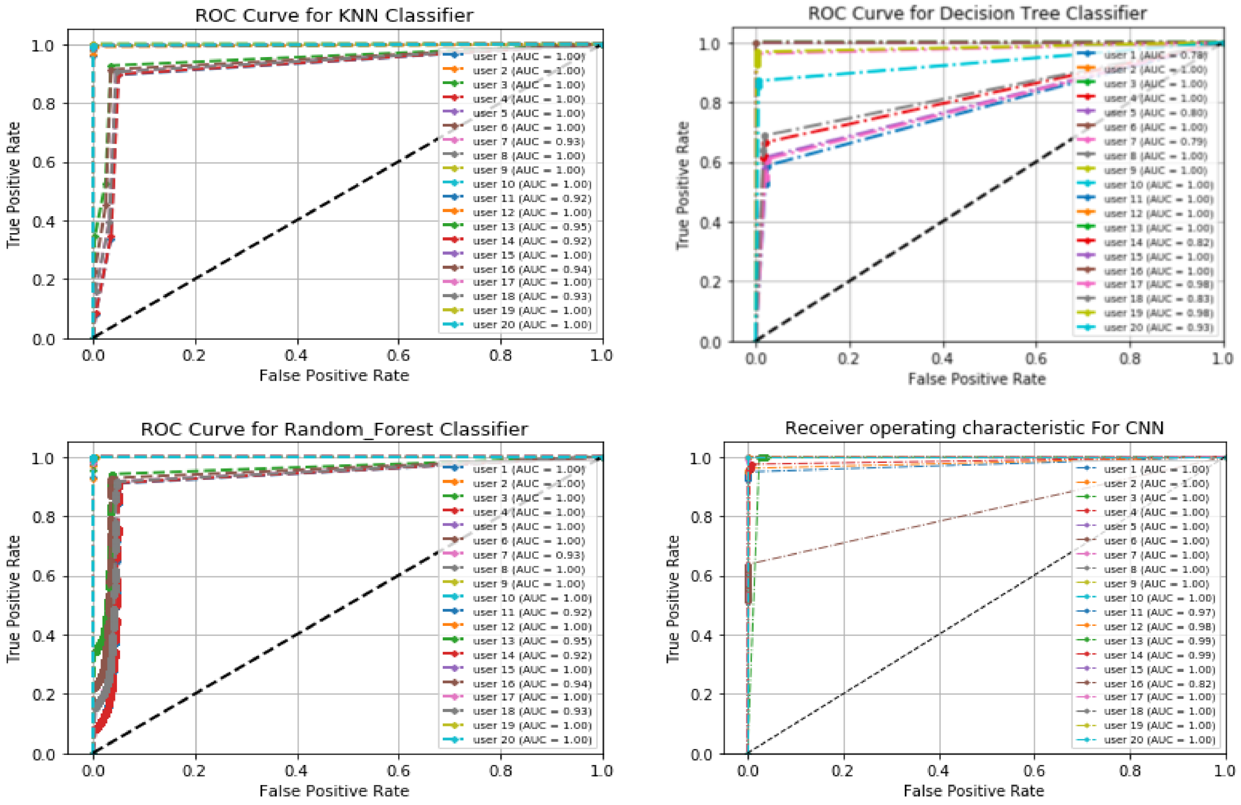


Figure 12. ROC curves for continuous authentication—single mouse movement action.

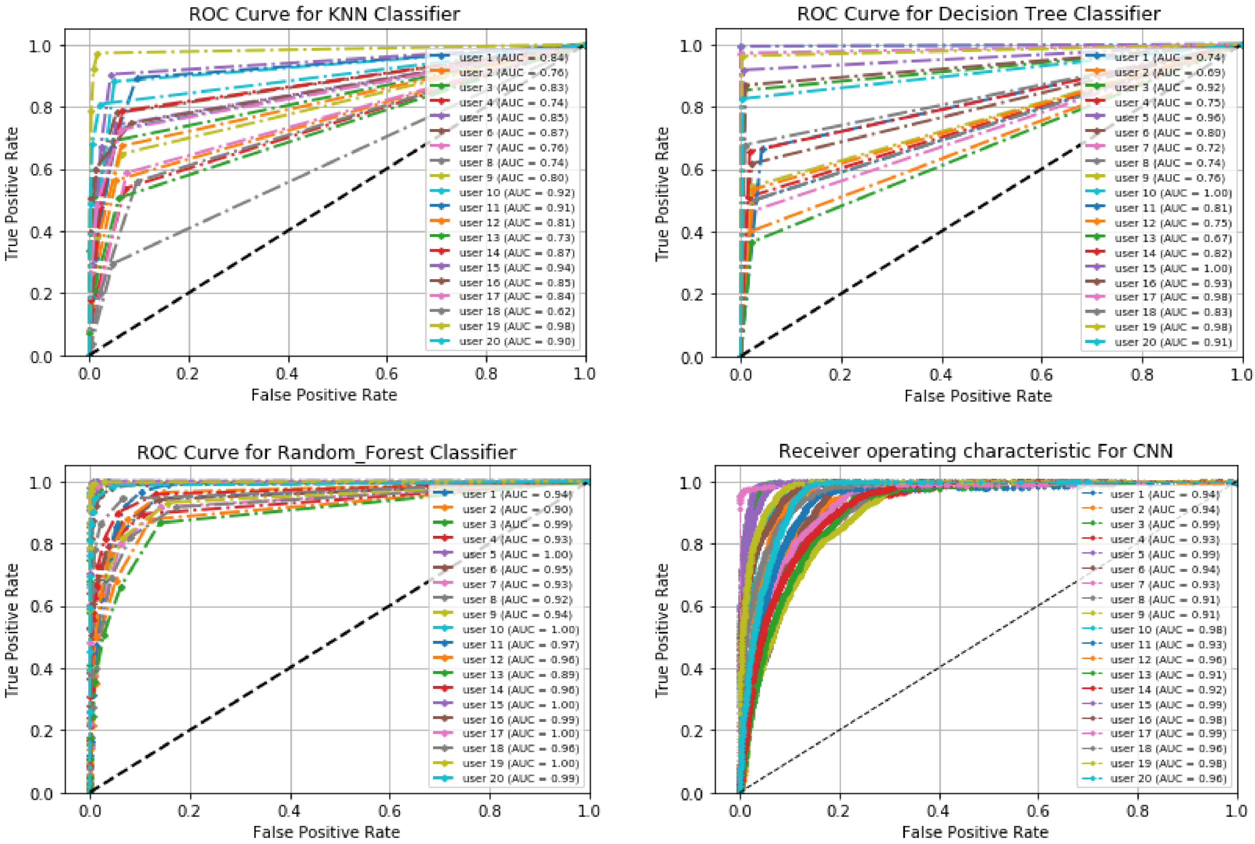


Figure 13. ROC curves for continuous authentication—single point-and-click action.

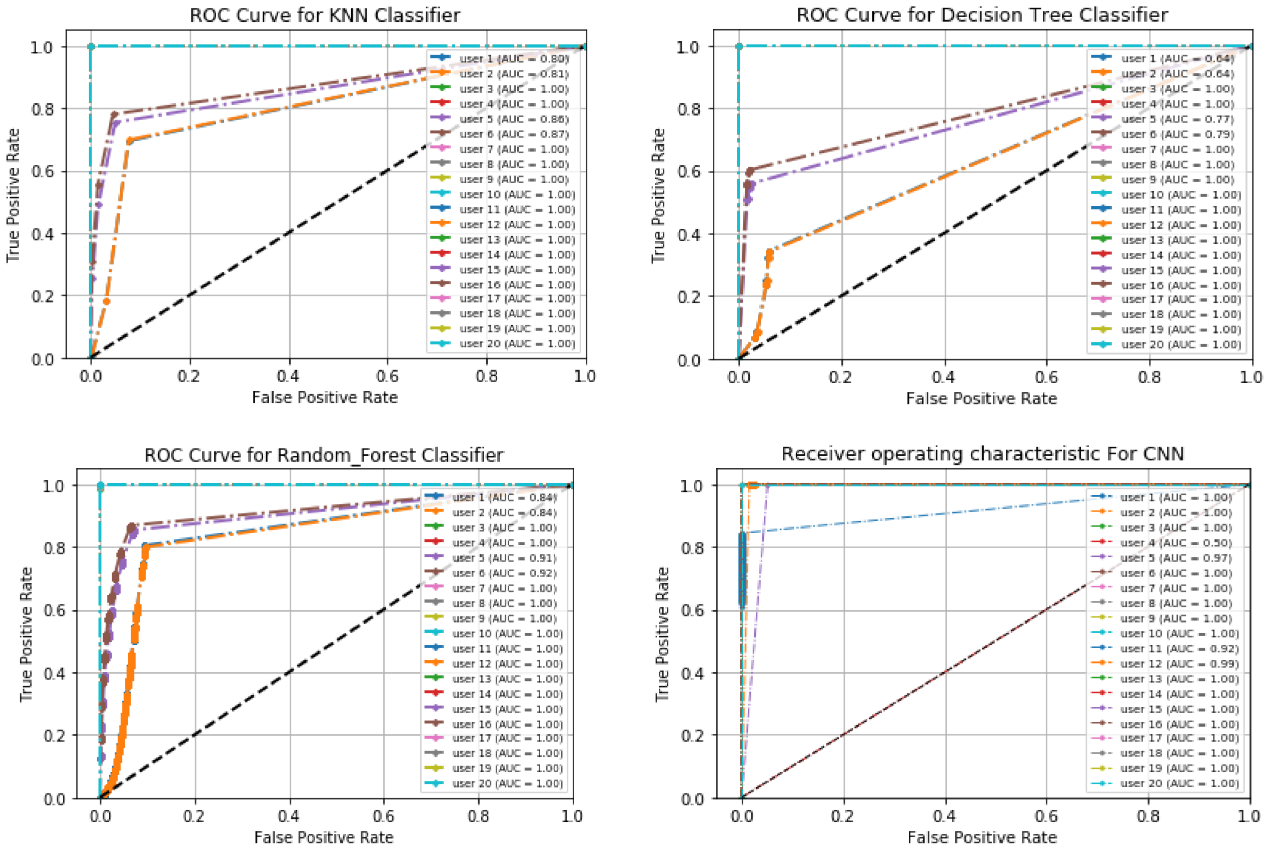


Figure 14. ROC curves for continuous authentication—set of mouse movement and point-and-click actions.

Table 12. AD evaluation—Scenario A (single mouse movement action): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.018%	0.020%	0.020%	0.052%
FRR	0.349%	0.389%	0.383%	0.990%
EER	0.045%	0.210%	0.035%	0.032%

Table 13. AD evaluation—Scenario B (a single point-and-click action): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.017%	0.018%	0.016%	0.050%
FRR	0.323%	0.335%	0.918%	0.906%
EER	0.101%	0.222%	0.065%	0.193%

Table 14. AD evaluation—Scenario C (set of mouse movement and point-and-click actions): FAR, FRR, and EER.

Classifier	KNN	DT	RF	CNN
FAR	0.026%	0.020%	0.029%	0.050%
FRR	0.537%	0.603%	0.510%	0.945%
EER	0.163%	0.349%	0.111%	0.234%

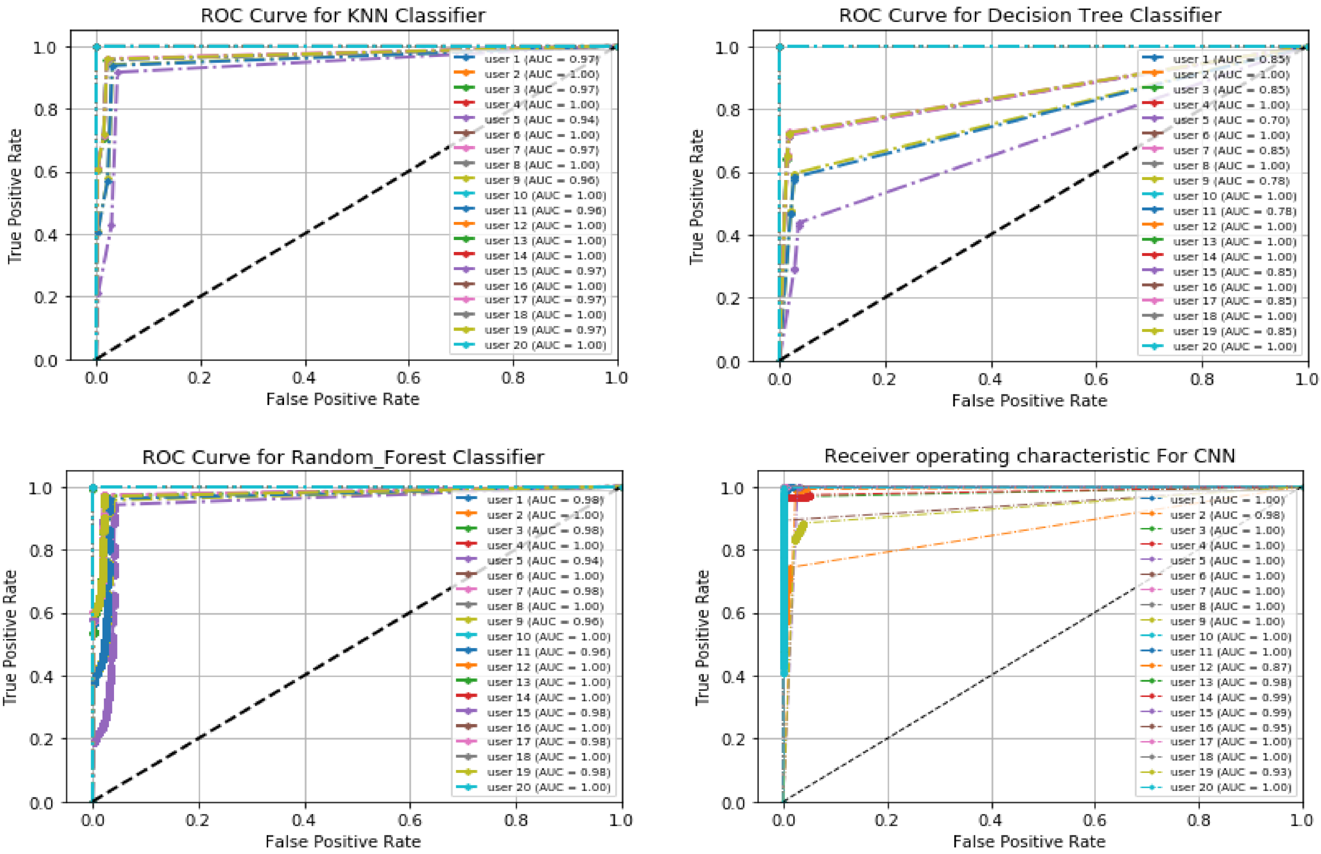


Figure 15. ROC curves for anomaly detection—single mouse movement action.

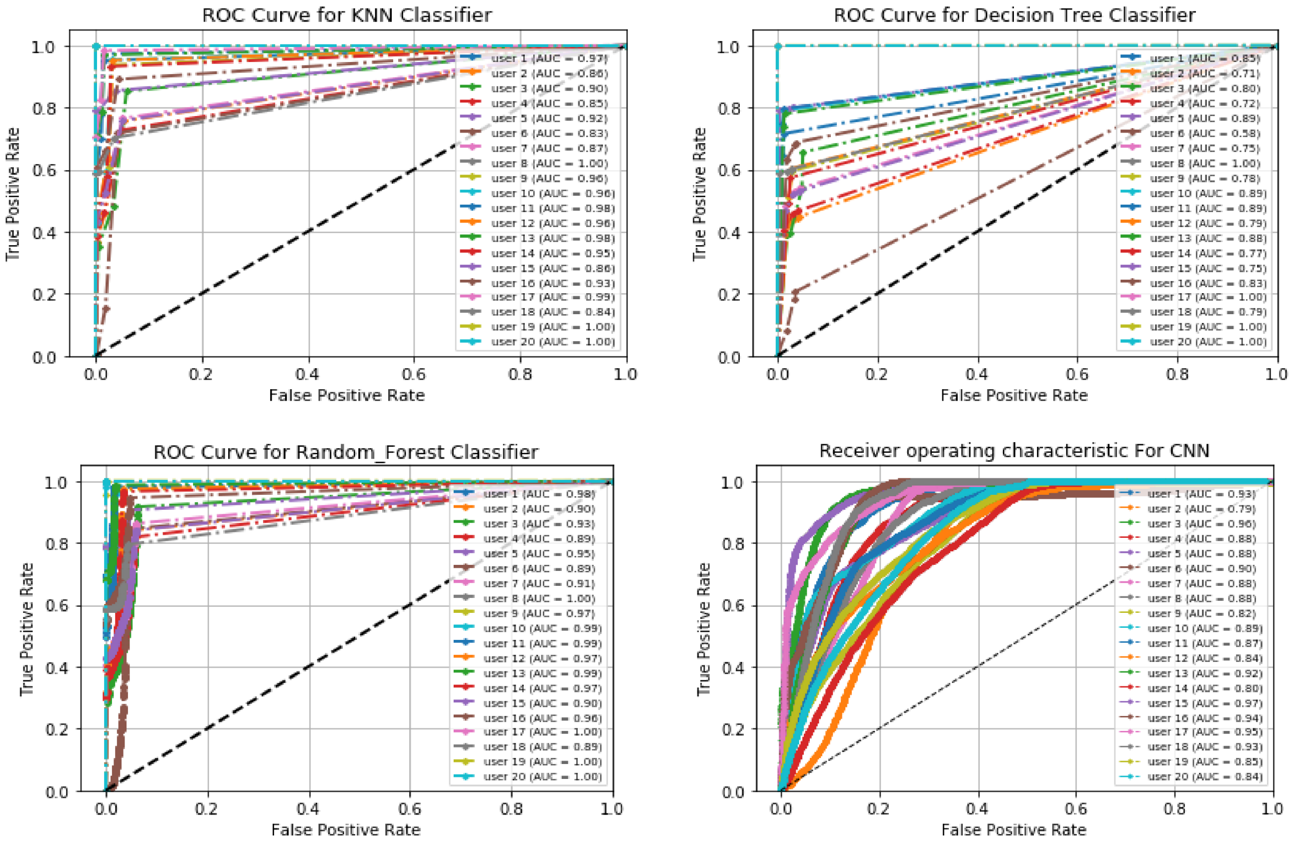


Figure 16. ROC curves for anomaly detection—single point-and-click action.

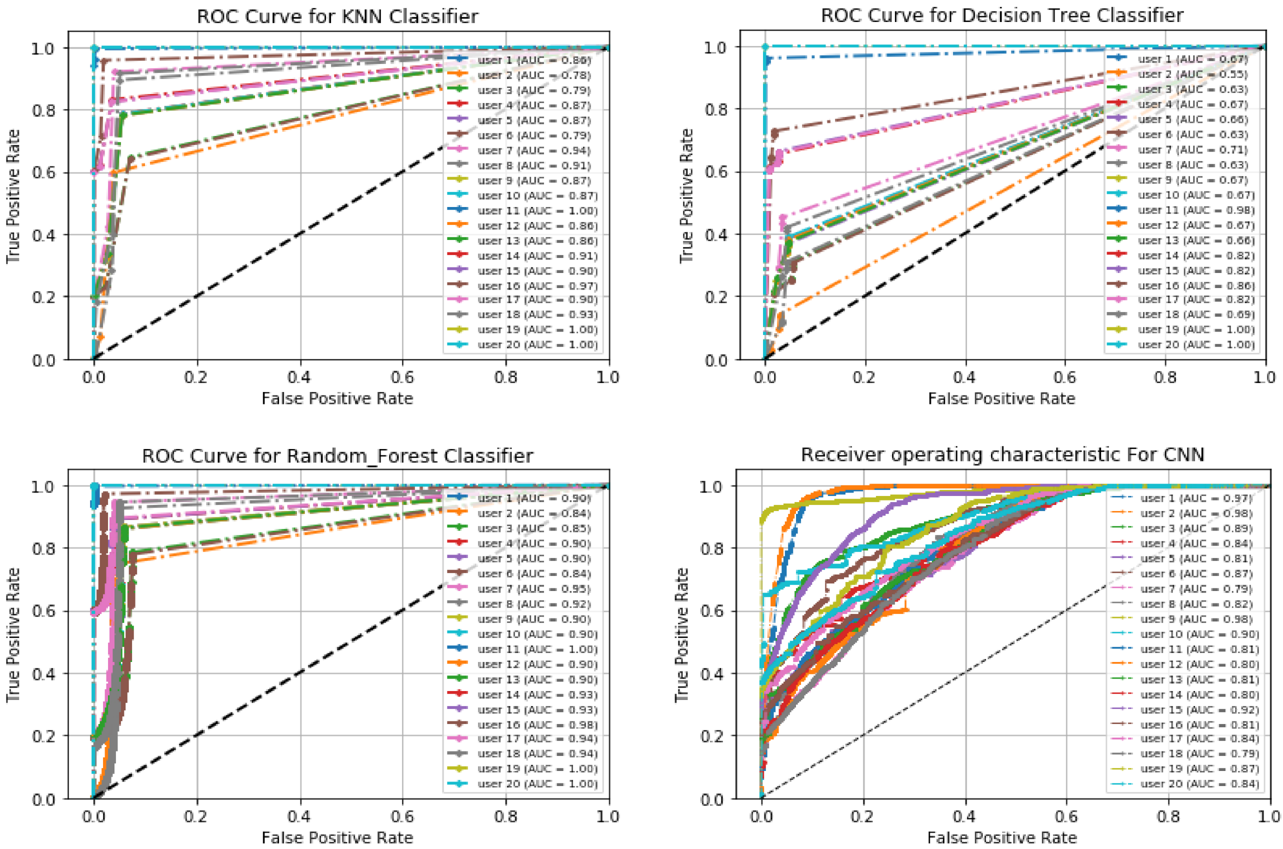


Figure 17. ROC curves for anomaly detection—set of mouse movement and point-and-click actions.

6.3. Comparison with the State-of-the-Art

Antal and Egyed-Zsigmond [22] conducted an evaluation study for impostor detection using the Balabit dataset that contains only 10 users [23]. They extracted 39 features and used a random forest classifier with an average accuracy of 81.17%. Salman and Hameed [38] investigated the performance of a continuous authentication system using a Gaussian naive Bayes classifier. They conducted their experiments using the dataset of Ahmed and Traore [40], consisting of eight features for each user data, and obtained an accuracy of 93.563%. Compared to these two works, we have presented an empirical evaluation of online CA and AD using 87 features extracted from the raw mouse dataset. For each of the CA phase and the AD phase, three scenarios were evaluated: Scenario A (single mouse movement action), Scenario B (single point-and-click action), and Scenario C (set of mouse movement and point-and-click actions). In both phases, KNN, DT, RF, and CNN classifiers were used separately to identify users by their mouse behavior; the evaluation assessed the ability of each classifier to distinguish between genuine users and imposters.

7. Conclusions

Mouse dynamics are behavioral biometrics that can be applied in different security fields such as human identification. This study was performed using the mouse dynamics data collected in the CIB Laboratory. Data collection involved 20 users with restriction of environment. Each session's data were segmented into two types of mouse actions: mouse movement actions and point-and-click actions. The results prove the capability of the proposed approaches to differentiate a legitimate user from an illegitimate user. We studied the efficiency of CA and AD using different ML and DL algorithms. For user identification, we considered three scenarios: Scenario A, a single mouse movement action; Scenario B, a single point-and-click action; and Scenario C, a set of mouse movement and point-and-click actions. The proposed CNN model showed a promising performance on our dataset. The performance of the CNN was also compared with the traditional ML algorithms. The results indicate that our approach can differentiate an authentic user from a fraudulent user with a comparatively high degree of accuracy.

Author Contributions: Conceptualization, S.A. and K.R.; methodology, S.A. and K.R.; software, S.A.; validation, S.A. and N.A.; formal analysis, S.A., N.A., and K.R.; investigation, S.A., N.A. and K.R.; resources, S.A. and N.A.; data curation, S.A. and K.R.; writing—original draft preparation, S.A.; writing—review and editing, K.R.; visualization, S.A.; supervision, K.R.; project administration, K.R.; funding acquisition, K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

Institutional Review Board Statement: The Cyber Identity and Biometrics Lab: Mouse Dynamics Data Collection has been reviewed by the Institutional Review Board (IRB), NC A&T Division of Research and Economic Development, North Carolina A&T State University and was determined to be exempt from further review. IRB# 19-0105.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahmed, A.A.E.; Traore, I. A New Biometric Technology Based on Mouse Dynamics. *IEEE Trans. Dependable Secur. Comput.* **2007**, *4*, 165–179. [CrossRef]
2. Chiasson, S.; Forget, A.; Stobert, E.; van Oorschot, P.C.; Biddle, R. Multiple Password Interference in Text and Click-Based Graphical Passwords. 11. In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09), Chicago, IL, USA, 9–13 November 2009; pp. 500–511.
3. Chong, P.; Elovici, Y.; Binder, A. User Authentication Based on Mouse Dynamics Using Deep Neural Networks: A Comprehensive Study. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1086–1101. [CrossRef]
4. Enström, O. Authentication Using Deep Learning on User Generated Mouse Movement Images. 2019. Available online: http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-74203 (accessed on 8 April 2021).
5. Guglielmo, L.; Geiger, S.; Burns, C.; Bondalapati, R.; Sidaras-Tirrito, M. Using Mouse Movement Biometrics to Authenticate Students Taking Online Multiple-Choice Exams. 7. Available online: https://www.semanticscholar.org/paper/Using-Mouse-Movement-Biometrics-to-Authenticate-Guglielmo-Geiger/c41a609c5eb3f5a53cfcfa9ec4250c7e24dda999 (accessed on 8 April 2021).
6. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
7. Wolf, T.; Babaee, M.; Rigoll, G. Multi-view gait recognition using 3D convolutional neural networks. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 4165–4169. [CrossRef]
8. Ahmed, A.A.E.; Traore, I. Anomaly intrusion detection based on biometrics. In Proceedings of the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop, West Point, NY, USA, 31 March–1 April 2005; pp. 452–453. [CrossRef]
9. Chudá, D.; Krátky, P. Usage of computer mouse characteristics for identification in web browsing. In Proceedings of the 15th International Conference on Computer Systems and Technologies-CompSysTech'14, Ruse, Bulgaria, 27–28 June 2014; pp. 218–225. [CrossRef]
10. Yampolskiy, R.V. Human Computer Interaction Based Intrusion Detection. In Proceedings of the Fourth International Conference on Information Technology (ITNG'07), Las Vegas, NV, USA, 2–4 April 2007; pp. 837–842. [CrossRef]
11. Passerini, E.; Paleari, R.; Martignoni, L. How Good Are Malware Detectors at Remediating Infected Systems? In *Detection of Intrusions and Malware, and Vulnerability Assessment*; Flegel, U., Bruschi, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 21–37. [CrossRef]
12. Shen, C.; Cai, Z.; Guan, X.; Du, Y.; Maxion, R.A. User Authentication Through Mouse Dynamics. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 16–30. [CrossRef]
13. Chauhan, V.; Pilaniya, A.; Middha, V.; Gupta, A.; Bana, U.; Prasad, B.R.; Agarwal, S. Anomalous behavior detection in social networking. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–5. [CrossRef]

14. Jain, A.K.; Hong, L.; Pankanti, S.; Bolle, R. An identity-authentication system using fingerprints. *Proc. IEEE* **1997**, *85*, 1365–1388. [CrossRef]
15. Liu, S.; Silverman, M. A practical guide to biometric security technology. *IT Prof.* **2001**, *3*, 27–32. [CrossRef]
16. Chuda, D.; Kratky, P.; Tvarozek, J. Mouse Clicks Can Recognize Web Page Visitors! In Proceedings of the 24th International Conference on World Wide Web-WWW '15 Companion, Florence, Italy, 18–22 May 2015; pp. 21–22. [CrossRef]
17. Hamid, N.A.; Safei, S.; Satar, S.D.M.; Chuprat, S.; Ahmad, R. Mouse movement behavioral biometric systems. In Proceedings of the International Conference on User Science and Engineering (i-USEr), Selangor, Malaysia, 29 November–1 December 2011; pp. 206–211. [CrossRef]
18. Hashia, S.; Pollett, C.; Stamp, M. On Using Mouse Movements as a Biometric. 5. In Proceedings of the International Conference on Computer Science and its Applications, Singapore, 9–12 May 2005.
19. Gamboa, H.; Fred, A. A Behavioural Biometric System Based on Human Computer Interaction. *Proc. SPIE* **2004**, *5404*, 381–392. [CrossRef]
20. Pusara, M.; Brodley, C.E. User re-authentication via mouse movements. In Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security-VizSEC/DMSEC '04, Washington, DC, USA, 29 October 2004. [CrossRef]
21. Ahmed, A.A.E.; Traore, I. Dynamic sample size detection in continuous authentication using sequential sampling. In Proceedings of the 27th Annual Computer Security Applications Conference, Orlando, FL, USA, 5–9 December 2011; pp. 169–176. [CrossRef]
22. Antal, M.; Egyed-Zsigmond, E. Intrusion detection using mouse dynamics. *IET Biom.* **2019**, *8*, 285–294. [CrossRef]
23. Fülöp, Á.; Kovács, L.; Kurics, T.; Windhager-Pokol, E. Balabit Mouse Dynamics Challenge Data Set. 2016. Available online: https://medium.com/balabit-unsupervised/releasing-the-balabit-mouse-dynamics-challenge-data-set-a15a016fba6c (accessed on 8 May 2021).
24. Tan, Y.X.M.; Iacovazzi, A.; Homoliak, I.; Elovici, Y.; Binder, A. Adversarial attacks on remote user authentication using behavioural mouse dynamics. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–10.
25. da Silva, V.R.; Costa-Abreu, M.D. An empirical biometric-based study for user identification with different neural networks in the online game League of Legends. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6. [CrossRef]
26. Zheng, N.; Paloski, A.; Wang, H. An Efficient User Verification System Using Angle-Based Mouse Movement Biometrics. *ACM Trans. Inf. Syst. Secur.* **2016**, *18*, 1–27. [CrossRef]
27. Feher, C.; Elovici, Y.; Moskovitch, R.; Rokach, L.; Schclar, A. User identity verification via mouse dynamics. *Inf. Sci.* **2012**, *201*, 19–36. [CrossRef]
28. Sayed, B.; Traore, I.; Woungang, I.; Obaidat, M.S. Biometric Authentication Using Mouse Gesture Dynamics. *IEEE Syst. J.* **2013**, *7*, 262–274. [CrossRef]
29. Shen, C.; Cai, Z.; Guan, X.; Maxion, R. Performance evaluation of anomaly-detection algorithms for mouse dynamics. *Comput. Secur.* **2014**, *45*, 156–171. [CrossRef]
30. Zheng, N.; Paloski, A.; Wang, H. An efficient user verification system via mouse movements. In Proceedings of the 18th ACM Conference on Computer and Communications Security-CCS '11, Chicago, IL USA, 17 October 2011; pp. 1–27. [CrossRef]
31. Shen, C.; Cai, Z.; Guan, X. Continuous authentication for mouse dynamics: A pattern-growth approach. In Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), Boston, MA, USA, 25–28 June 2012; pp. 1–12. [CrossRef]
32. Bailey, K.O.; Okolica, J.S.; Peterson, G.L. User identification and authentication using multi-modal behavioral biometrics. *Comput. Secur.* **2014**, *43*, 77–89. [CrossRef]
33. Mondal, S.; Bours, P. Continuous Authentication in a real world settings. In Proceedings of the 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), Kolkata, India, 4–7 January 2015; pp. 1–6. [CrossRef]
34. pyHook. 2021. Available online: https://pypi.org/project/pyHook/ (accessed on 8 April 2021).
35. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
36. Jovic, A.; Brkic, K.; Bogunovic, N. An overview of free software tools for general data mining. In Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 26–30 May 2014; pp. 1112–1117. [CrossRef]
37. Almalki, S.; Chatterjee, P.; Roy, K. Continuous Authentication Using Mouse Clickstream Data Analysis. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage*; Wang, G., Feng, J., Bhuiyan, M.Z.A., Lu, R., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; Volume 11637, pp. 76–85. [CrossRef]
38. Salman, O.A.; Hameed, S.M. Using Mouse Dynamics for Continuous User Authentication. In Proceedings of the Future Technologies Conference (FTC) 2018; Arai, K., Bhatia, R., Kapoor, S., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; Volume 880, pp. 776–787. [CrossRef]
39. Damousis, I.G.; Argyropoulos, S. Four Machine Learning Algorithms for Biometrics Fusion: A Comparative Study. *Appl. Comput. Intell. Soft Comput.* **2012**, *2012*, 242401. [CrossRef]
40. Ahmed, A.A.E.; Traore, I. *Mouse Dynamics Biometric Technology; Behavioral Biometrics for Human Identification: Intelligent Applications*; IGI Global: Hershey, PA, USA, 2010. [CrossRef]