User Behavior Continuous Authentication in Secure Transactional Application

Daniel-Marian DĂNILĂ[1], Natalia Sierra TOLEDO[2], Alin ZAMFIROIU[1,2,3]
[1]Bucharest University of Economic Studies, Romania
[2]Utah Tech University, Saint George, Utah, USA
[3]National Institute for Research & Development in Informatics - ICI Bucharest, Romania
daniladaniel20@stud.ase.ro, natalia.sierra.toledo@utahtech.edu, alin.zamfiroiu@csie.ase.ro

The purpose of this paper is to find the appropriate physical and behavioral biometric as well as contextual indicators for continuous authentication in a secure transactional application. The indicators must have good accuracy in identifying a user and be able to form the basis of an anomaly detector. In addition, their combination must maintain the usability of the application. Therefore, a compromise between accuracy and usability will be made so that the user can use the application easily and at the same time be protected from a potential thief. The paper relies on research and results from over twenty-five articles in the specialized literature to identify the best indicators, classifiers, and algorithms that have demonstrated good accuracy and usability. Furthermore, additional indicators specific to a banking application were considered. The identified indicators will be collected during the use of the application from several volunteer users and subsequently analyzed to obtain the results. To decide whether a user is legitimate, a score is calculated for the current session based on the indicators that have a weight based on the importance that was established following an opinion questionnaire. If the score falls below a certain value, then the current session is stopped immediately, and the person is logged out. The study was conducted on a banking application and among the indicators are the preference for displaying or hiding the balance, dominant hand for using the application, the way in which they return to the application, either through the PIN number or by using the fingerprint, the location and time from which the application is accessed, the amounts transacted, the way in which they press buttons. Overall, this research paper contributes to the literature by identifying and testing a unique set of indicators suitable for continuous authentication using behavioral biometric data in the context of a banking application.
Keywords: *Behavioral, Biometrics, Continuous, Authentication, Transactional*
DOI: 10.24818/issn14531305/30.1.2026.03

1 Introduction

Continuous authentication has been extensively studied in recent years. However, few studies have explored it on a transactional application. This study was designed to discover the relevant indicators that best differentiate user behaviors in a banking application so that it can guarantee that an authenticated person is who he claims to be.

The relevance of this topic is because in the past decade, the number of smartphone users has increased, from 2.5 billion in 2016 [1] to 5.2 billion in 2025 [2] which makes the smartphones sold more than the laptops worldwide [3]. Besides that, mobile phones now have compute and storage capabilities comparable to a personal computer [4]. After wallet and keys, smartphones are checked before leaving the home [5] since mobile phones are used daily for important tasks and store private, sensitive and vital information related to banking, medical or, for convenience purposes, application account credentials [3], [5], [6], [7], [8]. This creates a demand for additional security features on the devices to keep the data safe in case of theft or lost [3], [4], [6], [7] as smartphones are easy to steal because of their small size [5].

Compromised accounts are daily events and cyberattacks happen every year due to poor authentication mechanisms which, in general, are based on username and password [9]. One of the reasons is related to weak short passwords which can be broken easily.

Furthermore, strong long passwords are forgettable, wherefore users tend to avoid them [10]. People tend to prefer to have simple passwords or PIN such as birthdates, names, phones, addresses as they are easier to remember. The downsize is the vulnerability to dictionary attacks which lead to unauthorized access [12]. Around 40% of the users have a 4-digit PIN which can be broken by an impostor in less than three attempts in 9.23% of cases [13]. Moreover, around 10% of people claim they do not update their operating system or device application [5].

Traditional authentication, also called knowledge-based or static authentication, (e.g. username and password, PIN) has the disadvantage of offering the same level of security to any application, meaning a financial application has the same level of protection as an instant messaging application [4]. Regardless of how good static authentication implementation is, they remain vulnerable to situations where a user leaves their device open and unattended. This represents a perfect opportunity for an impostor to gain unauthorized access to data [10]. In addition, knowledge-based authentication is vulnerable to social engineering [11]. The way the attackers can be stopped is through continuous authentication that re-verifies the user's identity [10].

Continuous authentication (CA) is seen as a support process to the traditional authentication mechanisms [4] which validate user identity without his active intervention [14]. It is based on physical and behavioral biometrics recognition [5], [9], [10], [15]. Physical characteristics are considered static features because they do not change over time: fingerprints, voice, face, teeth, hand, iris, vein patterns, nervous system signals from electroencephalography (ECG), electrocardiography (EEG) etc. [10], [15], [16], [17]. While behavioral are ones related to the way a person reacts to an event: voice, signature speed and shape, motion dynamics (e.g. walking and gait), eye movement, keyboard and mouse usage dynamics, keystroke dynamics, touch gestures (e.g. taps, finger pressure, finger touch duration, touch width, hold time, inter-key actions, swipes, slides, handwriting, flicks, how the phone is grasp and hold, and multi-finger input for multi-touch screens), hand waving, stylometry dynamics, linguistic profiling etc [12], [10], [15], [16], [17]. Physical biometric authentication has a high accuracy and acceptance from the users compared to the behavioral one. However, it is also more expensive and is vulnerable to cases when user leave their device open and unsupervised [4], [15].

Generally speaking, biometrics recognition is safer than traditional authentication as they are harder to be lost, stolen, copied or forged [1]. Also, there are no two users which use the same set of applications at the same time of the day and in the same way [6].

Based on the results of this paper, a software developer or an organization may decide to implement this solution in its own application to increase its security and reduce the phenomenon of electronic fraud.

2 Literature review

Authentication provides access into a system and grants specific permissions, rights and privileges based on the identity claimed [18]. There are five types of authentications: based on what you know (e.g. passwords, PIN), based on what you have (e.g. devices, tokens, smartcards), based on what you are (e.g. physical biometric), based on how you react (e.g. behavioral biometric) and based on context aware factors (e.g. application used, physical location, date and time, IP address, Global Positioning System (GPS) data, wireless communication such as Wi-Fi and Bluetooth, device specific data such as operating system, battery usage, power consumption, network usage and web browsing history) [17], [18], [19], [20].

Two or multi-factor or multimodal authentication is a combination of these factors (e.g. physical biometric with behavioral biometric, biometric with non-biometric) which builds additional layers of security and makes it harder for attackers to break through [6], [12], [17], [19]. Additionally, it provides better accuracy in correctly identifying a user when adding appropriate methods together while it can also decrease accuracy if a characteristic with unrepresentative data is used [4], [5].

However, user convenience and application usability are also important, and the combination of authentication methods must consider a trade-off between usability and accuracy [12], [19]. Unimodal authentication is insecure because it relies solely on the traits of a characteristic that can be erroneously collected for reasons such as changes in both physical and emotional behavior of the user [4].

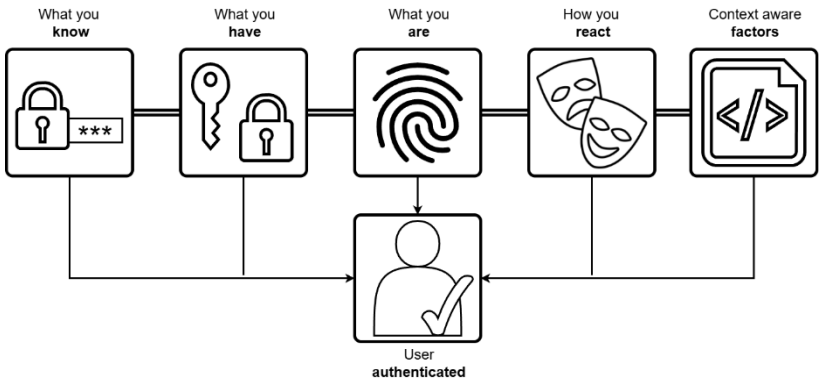

Fig. 1. Authentication methods

The five authentication methods can be subclassified into two categories: static (e.g. knowledge-based, possession-based, physical-based biometric) which checks user identify only once at the beginning and continuous (e.g. behavioral-based biometric, context-based) which verify a person's identity dynamically while they are using the device [13]. There are no international standards (e.g. ISO) for continuous authentication [10].

Current security mechanism of smartphones after being idle for a few seconds to minutes is to lock itself and request again the user to authenticate using a static method such as knowledge-based (e.g. PIN number or passphrase) or physical-based biometric (e.g. face or fingerprint) [13]. The downside of this flow is that the knowledge-based authentication is vulnerable to theft, shoulder-surfing and tracking user's finger smudge which makes it inefficient as it can easily grant a criminal access to sensitive personal data and financial applications [17]. Furthermore, it is vulnerable to side-channels attacks (e.g. video recordings, system timing information, motion sensors and wireless signals) or thermal imaging technology [12], [13], [14]. Even physical-based biometric authentication is not completely secure as it is exposed to smudge, counterfeit and replay attacks [10], [13].

Physical biometrics are considered a secure way of authentication because of the unique characteristics of a human and are more convenient for the user as there are no secrets to memorize. Additionally, the theft of these properties is harder compared to knowledge-based authentication. Although, once stolen the use of physical biometric is forever insecure as they do not change. Another disadvantage of this method would be the need for dedicated sensors which some smartphones do not possess [12].

Behavioral-based biometric authentication includes attacks like brute force, observation (e.g. shoulder surfing, video recording, photo shooting), impersonation (e.g. replay, imitation and synthesis attack) and side-channel (e.g. system timing information, motion sensors and wireless signals). This type of authentication is the safest against brute force and observation attacks because it is more abstract and thus, harder to forge [12]. This imply that attackers put in more effort to bypass it as it is context based [19].

To protect to the mentioned attacks from above, a new model must be used, such as behavioral biometrics continuous authentication (BBCA) which is based on how the user react to external events. Biometrics were primarily used for initial authentication and was found to be insufficient as an application is still vulnerable to subsequent attacks. For this reason, BBCA is proposed as it re-authenticates the user identity during a session [17].

Continuous authentication is a permanent, implicit, passive and progressive way to monitor, recognize and identify users by their features and actions to prevent unauthorized access [3], [10]. User verification in is performed periodically, and it compares the result with a predefined threshold [4].

Behavioral biometrics authentication is considered reliable, convenient, trusted and with a high degree of usability by the users [7]. Furthermore, they are less intrusive than physical biometrics and user tend to prefer them as they are over less private information about them. They are also repetitive which makes them appropriate for continuous authentication [12]. The goal of both methods is to provide identification and authentication for user recognition. Collection, extraction, comparison and match or mismatch of biometric data are the key four steps in user authentication based on user behavior [15].

Psychological biometrics have higher accuracy compared to behavioral biometrics and context-aware methods. In general, the latter do not reach 100% accuracy, which means that there is a risk of false negatives or false positives [10]. Moreover, user behavior and habits change over time for different reasons, and this must be analyzed in the informatic system using continuous learning techniques. Furthermore, injuries or a panic situation instantly changes behavior [3], [17].

Mobile devices upgraded their computational and storage capabilities which make them more suitable for continuous authentication. Furthermore, there are more and better advanced built-in sensors in today's smartphones which facilitate the collection of behavioral biometrics [1], [4], [16]. Examples include motion (e.g. accelerometer, gyroscopes, magnetometer), environmental (e.g. sound, light, internal temperature), position (e.g. GPS, Wi-Fi, proximity) and touch (e.g. capacitive, multi-touch, fingerprint) sensors [1], [4], [5], [21]. It is also important to consider requesting user permission to accept different sensor access when collecting behavioral data. In absence of them, the identification and continuous authentication deteriorate in terms of accuracy [16]. Beyond this, there are four types of data subclassification that sensors can collect: application dependent, application independent, device dependent and device independent. It is important to make this categorization because some user behaviors are not the same on different devices and applications [16].

Authentication systems make the decision to allow access either based on the score of individual behavioral traits or the score resulting from their fusion. The former has better accuracy. The performance is calculated with the help of confusion matrix (e.g. True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN)) and a threshold is defined for classifying genuine or impostor users [17].

Table 1. Confusion Matrix

	Actual Positive	**Actual Negative**
Predicted Positive	True Positives (TP)	False Positives (FP)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

There are multiple indicators used to measure the performance of the methods used for behavioral continuous authentication. A few notable examples include False Rejection Rate (FRR), represents the share of legitimate users who are declined access, False Acceptance Rate (FAR), indicates the percentage of illegitimate user who are permitted access, Equal Error Rate (EER), the point where FRR is equal to FAR, Accuracy, quantifies the proportion of all predictions correctly labelled, Precision, captures the percentage of true positives among all positives and Recall, measures the percent of actual positives found [10], [14], [15].

False Rejection Rate (FRR) formula is the ratio between FN to the sum of FN and TP [10], [14], [15].

$$FRR = \frac{FN}{FN+TP} \quad (1)$$

False Acceptance Rate (FAR) formula is the ratio between FP to the sum of FP and TN [10], [14], [15].

$$FAR = \frac{FP}{FP+TN} \quad (2)$$

Equal Error Rate (EER) formula is when FRR and FAR curves intersect [1], [10].

$$EER = FRR = FAR \quad (3)$$

Accuracy formula is the ratio between TP and TN to the sum of TP, FP, TN and FN [5], [10], [14].

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

Precision formula is the ratio between TP to the sum of TP and FP [4], [5], [14].

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Recall formula is the ratio between TP to the sum of TP and FN [5] [14].

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

FRR, FAR and EER lower values means better results, while for Accuracy, Precision and Recall, higher values indicate good performance [1], [10].

Continuous authentication based on behavioral biometrics has the drawback of denying a legitimate user activity. As the mobile phone has imprecise sensors and environmental noise impact the accuracy, capturing data which changes over time such as weight, age and habits results in growth of FRR. Voice has the lowest FRR which led to be a popular method [12] along with finger pressure feature which alone has 99% accuracy [6] and it is harder to be copied by an imposter [17]. Moreover, using a digital glove increases the accuracy of touch gestures methods [17].

Anomaly detections work best for continuous authentication compared to binary or multiclass classifiers as it is impossible to have the impostor's data collected to distinguish between them and the owner of the smartphone [13]. Abnormal behavior includes different device settings (e.g. device screen size, language and model), situations where two consecutive user actions were performed from locations far apart, which would not have been physically possible for a person to reach by any means during the time involved [11].

In [17], some of the behavioral traits suggested for analysis are touch gestures, keystroke dynamics, location and time, handwaving, gait and voice.

2.1 Touch gestures

Biometric touch sensing integrates authentication into interaction. It collects touch sensors data [8]. They have the following features collected to uniquely identify a user: location points (x, y), time stamp, finger pressure on screen, finger blocked area, finger orientation, device orientation (e.g. portrait or landscape), touch duration, vibration and rotation [12], [17]. To further support the collection of these features, capacitive images are used. They are low quality fingerprints and are used to determine the orientation of fingers, the body part, palm touches, and hand poses. They do not require any additional hardware, sensors or anything else besides a smartphone which use capacitive touchscreen for detecting touches. Popular touch gestures are tapping, scrolling, and dragging [22].

The most accurate classifiers for this property are k-NN (k-Nearest Neighbor) and SVM (Support Vector Machine) with 99% precision [12], [17].

Touch gestures vulnerabilities include shoulder surfing attack, malware attacks, robotic attacks and generative attacks [10], [17].

2.2 Keystroke dynamics

Keyboard writing is a behavioral biometric which has risen in popularity in the past decade, and it consists of, among other things, analyzing a user typing rhythm and capabilities [1], [15]. It is determined by analyzing numerous events (e.g. key press, key hold, key release, key pressure when pressed and released, absolute or relative location to the device screen or to the button pressed, touch area of the finger, inter-time: release-press, inter-key: press-press, hold-time: press-release latencies, error rate measured by how many backspaces were pressed, distance in pixels between two buttons or touches, typing speed computed with Words Per Minute (WPM) or Characters Per Seconds (CPS) or Keystrokes Per Seconds (KPS) or adjusted WMP metrics, number of fingers used and finger drag) [9],

[14], [15], [17]. The authentication techniques use the time in milliseconds between two events or the time to write a group of characters called N-graph time (e.g. digraph, trigraph) [1], [9], [15]. This behavioral biometric, also known as keystroke, is spitted in two: static or fixed text (e.g. includes PIN numbers, username or password and passphrases) and dynamic or continuous or free text (e.g. one-time passwords and any unique text) [1], [15], [17]. Static keystroke means to test all users and extract their characteristics on the same predefined text. Dynamic keystroke means to continuous analyze the user input through the entire session of a digital platform [15]. Keyboard typing is considered unique [14].

According to [17], with an EER of 0.44% and 2.2%, SVM and the RF (Random Forest) classifier are most accurate classifiers.

Keystroke dynamics vulnerabilities include Frog-Boiling attack, Algorithmic attack, Mimic attack, Snoop-forge-replay attack the resist mimicry attack [10], [17].

2.3 Location and time

They are saved when initially launching the application and subsequent it periodically updates them. SVM with 1% EER and k-NN with 3% FAR reported the best accuracy [17].

2.4 Hand gestures

Hand gestures refer to in-air gestures or 3D gestures done by the user and can uniquely identify him. They are facilitated by the motion sensors of the mobile phone [12].

A subclass of hand gestures is hand waving which inspect the way a user interacts with his phone while holding it with the help of wrist twisting, speed, waving range and frequency. With not much research done on this metric, only RF classifier with 4% EER is considered acceptable [17].

In what follows, several implementations of behavioral continuous authentication will be presented accompanied with their system architecture or methodology.

In [5], deep neural networks are applied on large dataset, and the problem is defined as binary classification. Different modalities are test such as touch screen data only, motion sensors data only and both. 88% accuracy with 15% EER is attained.

In the [7] paper, to profile a user, the following characteristics were collected: application access timestamp, the action type (e.g. send, read etc.), message lengths, call durations and time between two consecutive events. The tested classifiers were Random Forest (RF), Support Vector Machine (SVM) and gradient boosting (GB) with GB having the lowest EER of 26.98%. For ranking the importance of the features and giving them weights, random forest algorithm was used to

In the recent study of [15], the algorithms used are intended to determine students' identity based on how they type and interact with the platform. The first algorithm is composed of two user profiles, both based on the user behavior within the application. The first profile is related to the predefined text recorded during authentication and the second profile is based on the activity within the platform during the session. The data necessary for this algorithm are collected from user authentication until the end of the session. The user profile is determined by how it uses the keyboard by analyzing the following characteristics: typing speed, pressing time, how the user deletes text, how the user selects text, how the user chooses to write capitalized letters, the position of the control keys that the user uses. Further, these values are used to create MED (Medium Euclidean Distance) which is calculated using Euclidean distances. The second algorithm takes into consideration the static text the username types at the authentication (username and password) and measures the writing duration for the first six diagraph. An average and standard deviation is calculated for both username and password. Further, a confidence interval is formed to determine user authenticity. The third algorithm is used when the user has an active session in the application, and it types the most common two-letter groups. It identifies the typing duration for those groups, and it uses the same statistical method from the second algorithm. The study achieved 3.57% FAR and 8.92% FRR.

[19] stands out by focusing on the proposed authentication scheme which is starting from unlocked to locked if a suspicious activity of user behavior is detected. The technology works in four steps. The first one is the start signal from where device sensors start to collect data until finalization. Then, as the second step, context recognition is carried out using convolutional neural network (CNN). At the third step, A-RNN model is used for processed modalities which are prepared signals from inputted signals. In the last step, the output of the previous model is processed with decision-making module. The result decides if user is authenticated or not. The application has the following attributes collected: type pattern (timing, touches location – keyboard hit-map and distance between touches and buttons center), swipe patterns, device small motions, app usage patterns, eye tracking and mobile grip pattern. 2.1% FAR and 3.9% FRR were achieved.

Some studies expressed recommendations for future research. Examples include that analysis for user's typing pattern on mobile phones has shown less interest compared to physical keyboard studies [1]. Additionally, most research on keystroke dynamics are conducted in English and leaves room for exploring situations when other languages are used, including their special letters. Also, the discriminative characteristics were not associated with the age, the current mood of the user or the current activity of the user (walking, lying, standing, sitting) which make room for further analysis [1].

3 Methodology

The aim of this study is to investigate the appropriate physical and behavioral biometric as well as contextual indicators for continuous authentication in a secure transactional application. For that, a mixed-methods approach was used. A quantitative questionnaire with 48 responses in which respondents were asked to quantify on a numerical scale from 1 to 10 how they assessed the behaviors mentioned as differentiating one user from another in the context of a banking application. The behaviors being location, time (date and time), navigation style and path, the way and style of using the keyboard to write, the way and style of pressing the screen with the finger(s), preference for using the left or right hand, authentication method (using fingerprint or access code), a user's preference to keep their balance visible or hidden while using the application, and amounts transacted. The results of the questionnaire can be seen in Figure 1.

The data obtained through the questionnaire was collected through the Google Forms application, the link of which was distributed to several groups, where the majority were students up to 30 years old. Few samples were collected from people over that age. The results of the form are used to assign percentages to individual properties to obtain a final aggregate result that takes everything into account. However, the form was designed at an early stage of this work to discover relevant properties, and it was later found that some overlapped. For example, touch events and left/right hand preference, which is why in the percentage distribution some characteristics were added together as one.

Qualitative interviews were also conducted through discussions with teachers, co-workers, and students to identify characteristics relevant to behavior-based continuous authentication. In addition, over 25 scientific articles from journals and conferences were analyzed to determine suitable characteristics and effective analysis methods for each type of behavioral indicator. Furthermore, to obtain results, the indicators described in the previous chapter were used: FRR, FAR, EER and Accuracy, to compare my results with other studies.

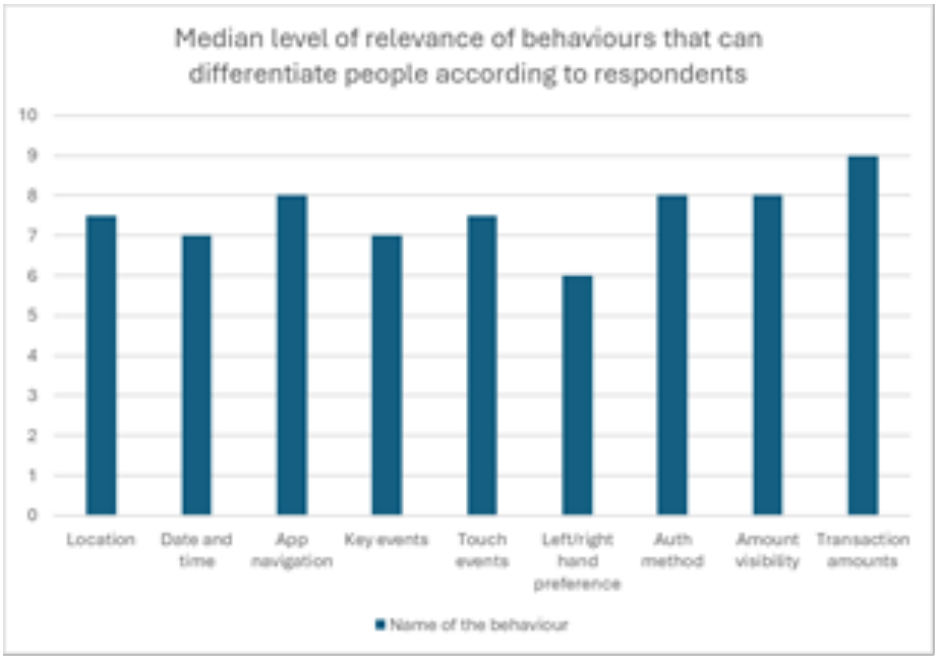

Fig. 2. Questionnaire results

Given that behavioral-based continuous authentication requires a lot of data to reach a meaningful conclusion, the HMOG behavioral dataset from College of William and Mary employees and students [46] was used in conjunction with my own collected data. Ethics was a priority factor in this analysis. Thus, I made sure that the authors of the dataset retrieved online explicitly highlighted that the data were collected with the consent of the respective individuals. Regarding the personal dataset, all individuals agreed to be part of the study anonymously. The sampling technique of the personal dataset was to choose appropriate individuals who showed interest in using the application over a period. The sample is not representative of the entire population as it only contains young people in the 20-24 age range.

The HMOG dataset [27] has 10 behavioral data tables, of which I kept only the ones relevant to my analysis: TouchEvent, KeyPressEvent, OneFingerTouchEvent, ScrollEvent, StrokeEvent, and Activity (since this table is referenced in the others and contains the user id). Also, some columns in the tables were deleted because their relevance was low. A notable example is the pressure with which a user presses the screen, although the property itself is very relevant, as several studies have shown, the values of this column were exclusively 1 and no differentiation between users could be made. Furthermore, each feature came in 2398 CSV files (multiplied by 6 tables used results in a total of 14388 files) with a different number of rows, implicitly with a different disk space occupied. Initially, I tried to insert all the properties into tables in the Oracle database, but I was encountered by the first limitation of this work, on the free version you can upload a maximum of 12 GB, which I exceeded and received the ORA-12954 error. However, even without this limitation, the time to insert all the data was expensive and long, which is why I limited myself to 500 files per feature, i.e. a total of 3000 files.

After putting the data into tables in the Oracle database, I started extracting features from each table using my own algorithms that I developed after observing the raw data. The properties common to all behaviors are the time expressed in sin and cos, the day of the week expressed in sin and cos, the month of the year expressed in sin and cos, whether the day of the week is on the weekend, the period of a day, the time difference between the screen press and release event, the time difference between two screen press events, the time difference between a screen release event and its press, and the phone orientation. Sin and cos were used for the aforementioned

properties for what's called encoding cyclical features [47], which helps machine learning models understand how certain properties work, like the hour within a day. Without this processing, the machine can't tell that 11 p.m. is closer to 1 a.m. than 1 a.m. is to 5 a.m.

The unique characteristics of the KeyPressEvent are the id of the keys used expressed in Android's KeyEvent constants, their occurrences, the total number of unique keys pressed, the total number of keys pressed, and the number of characters.

The common characteristics of the OneFingerTouchEvent, ScrollEvent, StrokeEvent, and TouchEvent behaviours are the starting and ending positions of the x and y coordinates, which quadrant of the screen they started and ended in, the individual distance between the starting and ending x points, and the same for y, the direction the x, y coordinates took, the Euclidean distance between the starting and ending x, y points, the angle created by the starting and ending x, y points, the average contact size made by the taps.

The unique feature of OneFingerTouchEvent is the number of movements performed by the second successive touch, the unique feature of TouchEvent is the number of movements performed with the first finger placed on the screen and the number of movements performed with the second finger placed on the screen, the unique features of ScrollEvent and StrokeEvent are the scrolling speed of the individual coordinates x, y and the magnitude speed.

In addition to the previously discussed features whose dataset was found online and used, the specific characteristics of the banking application are also added, such as amounts transacted, a user's preference to keep their balance visible or hidden while using the application, the authentication method (using fingerprint or access code), and the mode and path of navigation in the application.

After obtaining all the features, which were gradually thought out, added and tested, I moved on to the analysis part by using the classifiers that have proven the best results in previous works, namely k-NN, Random Forest and SVM. The performance indicator used in the first tests was accuracy, after which FAR, FRR and EER were also added.

4 Results

In this chapter, the results of the classifiers will be presented along with the parameters used and details about the features used.

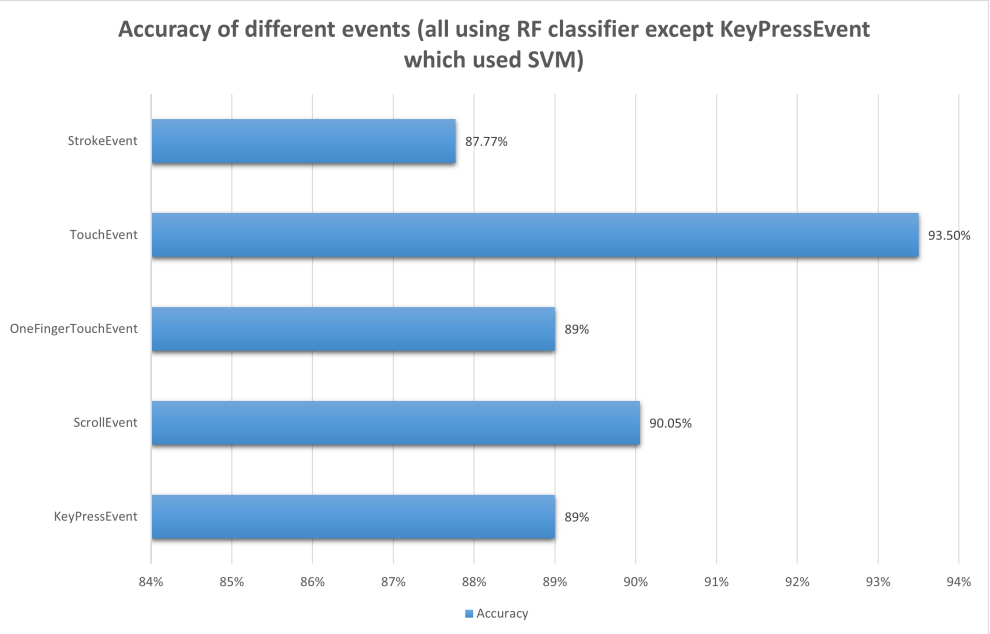

Fig. 3. Results visualized

The StrokeEvent's best results were obtained using RF with 87.77% accuracy, 0.74% FAR, 12.29% FRR and 2.72% EER, while the best results of the other classifiers were 79.64% accuracy, 1.12% FAR, 20.35% FRR and 12.71% EER for k-NN and 83.01% accuracy, 1.03% FAR, 16.98% FRR and 5.51% EER for SVM.

The way these results were obtained is as follows: (1) it was an early test and all the features were mainly raw data with little processing, (2) the way the data is processed was modified and in addition the properties of the duration between two successive presses and the duration between a release and a press were added, (3) the properties of the distance between the start and end of the individual points x, y were added, (4) the properties of the start and end screen quadrant of the coordinates x and y were added, and the direction of the coordinates were added, (5) the properties from (4) were one hot encoded, (6) one hot encoding was stopped and the magnitude speed property was added, (7) the same as (6) except that one hot encoding was started, (8) the average of the touch contact size, the Euclidean distance between the start and end points x, y, the angle made by the start and end points x, y were added, and removed the size of the start and end touch contact and the speed of x, y, (9) also deleted the phone orientation, the start and end screen quadrants and the direction, (10) also deleted the start and end coordinates x, y and their individual distances, (11) added the start and end coordinates x, y, (12) added the other previously deleted features, (13) added new time-related features, most of them after applying sin and cos to them: time, day of the week, day of the month, time of day, if it is a weekend, and enabled one hot encodings for direction, start and end screen quadrant, phone orientation and time of day, (14) like (13) but without one hot encoding, (15) deleted the previously mentioned time properties and added the raw value from which they were calculated: timestamp expressed in Unix Epoch Time in milliseconds.

The TouchEvent's best results were obtained using RF with 93.50% accuracy, 0.43% FAR, 6.49% FRR and 1.48% EER, while the best results of the other classifiers were 86.76% accuracy, 0.84% FAR, 13.23% FRR and 10.31% EER for k-NN and 91.97% accuracy, 0.5% FAR, 8.02% FRR and 3.42% EER for SVM.

The way these results were obtained is as follows: (1), (2) and (4) idem (1), (2) and (3) from StrokeEvent, (3) the properties of the number of movement actions, the average of the x and y coordinates, the average of the screen touch size were added and the characteristics related to the first and second individual touches were deleted, (5) the characteristics from (3) were deleted again and the time-related properties mentioned in StrokeEvent were added, the starting and ending screen quadrant of the x, y coordinates, the individual distance of the x, y coordinates, the direction of the x, y coordinates, the Euclidean distance of the starting and ending coordinates x, y, the angle made by the starting and ending coordinates x, y were also added.

The reason why the data in (4) is missing in RF and the data in (3) and (40) in SVM is due to the lack of information on this topic, most likely I forgot to record it in time when the scripts were executed. Also (1) and (2) of SVM have a question mark because I did not explicitly write the details of these tests, but I assumed that they fit given the order found in the notes.

The OneFingerTouchEvent's best results were obtained using RF with 89% accuracy, 0.56% FAR, 10.99% FRR and 2.17% EER, while the best results of the other classifiers were 79.02% accuracy, 1.06% FAR, 20.97% FRR and 4.64% EER for k-NN and 79.17% accuracy, 1% FAR, 20.82% FRR and 5.50% EER for SVM.

The way these results were obtained is as follows: (1) and (2) same as (1) and (2, 3) from StrokeEvent, (3) same as (5) from TouchEvent without the part where deletion is mentioned.

The data from (2) of the SVM model is missing due to the long execution waiting time for this behavior. The execution of (1) took around 10 hours and after this experience, for (3) and (4) the execution was no longer

awaited to calculate the best parameters, but some arbitrary ones were provided.

The ScrollEvent's best results were obtained using RF with 90.05% accuracy, 0.5% FAR, 9.46% FRR and 2.14% EER, while the best results of the other classifiers were 83.56% accuracy, 0.82% FAR, 16.43% FRR and 9.47% EER for k-NN and 88.87% accuracy, 0.58% FAR, 11.26% FRR and 3.86% EER for SVM. The way these results were obtained is as follows: (1) same as (1) from StrokeEvent, (2) same as (5) from TouchEvent, without the part where deletion is mentioned and in addition hot encodings were used, (3) same as (2) but without hot encodings.

The KeyPressEvent's best results were obtained using SVM with 89% accuracy, 0.57% FAR, 11% FRR and 4.70% EER, while the best results of the other classifiers were 88% accuracy, 0.63% FAR, 12% FRR and 6.25% EER for k-NN and 86% accuracy, 0.72% FAR, 14% FRR and 1.69% EER for RF.

The way these results were obtained is as follows: (1) same as (1) from StrokeEvent, (2) the time-related properties mentioned in StrokeEvent were added, the difference of between a press and a release, a press and the next press, a release and the next press, the number of unique keys pressed, the total number of keys pressed and the number of characters per second, (3) as (2) but with hot encodings.

5 Discussion

All models show a 20-30% increase in accuracy at one point, this is due to the new added properties, including those related to time. Until that moment, the timestamp was ignored and not introduced into the analysis because it generally decreased the quality of the model. However, after it was processed, the features resulting from it improved the model by 5-10%.

One thing that is noticeable is how the results generally have a low FAR and a high FRR. For a banking application, this is good news because it is preferable that an unauthorized user is not allowed to continue accessing the application. On the other hand, the usability of the application by legitimate users is somewhat neglected as the FRR reaches almost 13% in the case of a StrokEvent RF model, which performed the best of all for this type of behavior. A compromise can be achieved by using the EER metric, which increases the FAR and decreases the FRR compared to the previous situation.

To make future predictions, the models mentioned in the previous subchapter that performed best were chosen (Random Forest for StrokeEvent, TouchEvent, OneFingerTouchEvent and ScrollEvent, and SVM for KeyPressEvent) and to obtain an aggregated result, considering each behaviour, each individual result is assigned a weight obtained following the processing of the opinion questionnaire. My aggregate result is 89.94% accuracy, which is lower than some past works, but remains a solid base from which to improve.

6 Conclusions

User behavior continuous authentication in secure transactional applications was mainly carried out using Python technologies for everything related to data processing, analysis and prediction, Node.js and Dart & Flutter for the development of a banking application from which to collect data and test the prediction system created. To obtain a high level of accuracy, several continuous changes were made to the characteristics collected and analyzed to finally obtain the results highlighted in the paper, such as the aggregate accuracy of all behaviors of 89.94%. Furthermore, a good compromise between accuracy and usability was achieved, with FAR values of up to 2% and FRR values of up to 13% for the models used that gave the best results.

Therefore, I believe that the purpose of this paper has been fulfilled and that continuous authentication is essential in today's world given the vulnerabilities to which static authentication is exposed, also mentioned in the paper, such as leaving the mobile phone open unattended, social engineering, side channel attacks (e.g. covertly filming the moment the password is entered), shoulder surfing, theft etc. It is worth mentioning that static authentication must continue to exist, but together

with continuous authentication to benefit from the best security.

Regarding people's concerns about the protection of their personal data collected through continuous authentication, this is solved by storing behavioral data locally on the personal device, but this has the disadvantage of the high storage space of an application that uses such a system. Another disadvantage is that there are also times when continuous authentication fails because a person's behaviors change radically in certain situations of crisis, panic, or injury. Furthermore, behaviors naturally change over time, and if an application is rarely used, these changes will be detected as very different from the person's last records, which will result in the increase of FRR indicator.

In the future, this work can be improved by finding solutions to the previous mentioned drawbacks, exploring more behaviors and their extractable characteristics, obtaining more data from volunteer users over a longer period to test how the accuracy changes depending on the number of records collected and applying this system from the paper to a production application after the accuracy increases and remains stable at a percentage close to 100%.

References

[1] E. Maiorana, H. Kalita and P. Campisi, "Mobile keystroke dynamics for biometric recognition: An overview," *IET Biometrics*, vol. 10, no. 1, pp. 1-23, 19 December 2021.

[2] S. Gill, "How Many People Own Smartphones in the World? (2024-2029)," *Priori Data*, 1 January 2025. [Online]. Available: https://priori-data.com/data/smartphone-stats/. [Accessed 12 05 2025].

[3] A. Alzubaidi and J. Kalita, "Authentication of Smartphone Users Using Behavioral Biometrics," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1998-2026, 2016.

[4] M. Abuhamad, A. Abusnaina, D. Nyang and D. Mohaisen, "Sensor-Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65-84, 2021.

[5] H. Volaka, O. Basar, M. Isbilen and O. Incel, "Towards Continuous Authentication on Mobile Phones using Deep Learning Models," *Procedia Computer Science*, vol. 155, pp. 177-184, 2019.

[6] I. Stylios, O. Thanou, I. Androulidakis and E. Zaitseva, "A Review of Continuous Authentication Using Behavioral Biometrics," in *Proceedings of the SouthEast European Design Automation, Computer Engineering, Computer Networks and Social Media Conference*, Kastoria, Greece, 2016.

[7] S. Alotaibi, A. Alruban, S. Furnell and N. Clarke, "A Novel Behaviour Profiling Approach to Continuous Authentication for Mobile Applications," in *Proceedings of the 5th International Conference on Information Systems Security and Privacy* - ICISSP, Prague, 2019.

[8] C. Holz and M. Knaust, "Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication," in *UIST '15: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, Charlotte, NC, USA, 2015.

[9] T. Shimshon, R. Moskovitch, L. Rokach and Y. Elovici, "Continuous Verification Using Keystroke Dynamics," in *2010 International Conference on Computational Intelligence and Security*, Nanning, China, 2010.

[10] A. F. Baig and S. Eskeland, "Security, Privacy, and Usability in Continuous Authentication: A Survey," *Sensors*, vol. 21, no. 17, p. 5967, 2021.

[11] D. Preuveneers and W. Joosen, "SmartAuth: dynamic context fingerprinting for continuous user authentication," in *SAC '15: Proceedings of the 30th Annual ACM Symposium on Applied Computing*, Salamanca, Spain, 2015.

[12] W. Chen, W. Yan, C. Yingying, L. Hongbo and L. Jian, "User authentication on mobile devices: Approaches, threats

and trends," *Computer Networks*, vol. 170, p. 107118, 2020.

[13] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu and X. Zhou, "BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics," *Ad Hoc Networks*, vol. 84, pp. 9-18, 2019.

[14] L. de-Marcos, J.-J. Martínez-Herráiz, J. Junquera-Sánchez, C. Cilleruelo and C. Pages-Arévalo, "Comparing Machine Learning Classifiers for Continuous Authentication on *Mobile Devices by Keystroke Dynamics*," Electronics, vol. 10, no. 14, p. 1622, 2021.

[15] A. Zamfiroiu, D. Constantinescu, M. Zurini and C. Toma, "Secure Learning Management System Based on User Behavior," *Applied Sciences*, vol. 10, no. 21, p. 7730, 31 October 2020.

[16] X. Wang, T. Yu, M. Zeng and P. Tague, "XRec: Behavior-Based User Recognition Across Mobile Devices," Proceedings of the *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 26, 11 September 2017.

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

[18] K. A. Abu Bakar and G. R. Haron, "Adaptive Authentication based on Analysis of User," in *2014 Science and Information Conference*, London, 2014.

[19] D. Progonov, V. Cherniakova, P. Kolesnichenko and A. Oliynyk, "Behavior-based user authentication on mobile devices in various usage contexts," *EURASIP Journal on Information Security*, vol. 2022, no. 1, p. 6, 16 September 2022.

[20] I. Brosso, A. L. Neve, G. Bressan and W. V. Ruggiero, "A Continuous Authentication System Based on User Behavior Analysis," in *2010 International Conference on Availability, Reliability and Security*, Krakow, 2010.

[21] J. Ahn and R. Han, "Personalized Behavior Pattern Recognition and Unusual Event Detection for Mobile Users," *Mobile Information Systems*, vol. 9, no. 2, p. 360243, 2013.

[22] H. V. Le, S. Mayer and N. Henze, "Investigating the feasibility of finger identification on capacitive touchscreens using deep learning," in *IUI '19: Proceedings of the 24th International Conference on Intelligent User Interfaces*, Marina del Ray, California, 2019.

[23] R. Rieke, Z. Maria, R. Jürgen, G. Romain and G. Chrystel, "Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis," in *2013 International Conference on Availability, Reliability and Security*, Regensburg, 2013.

[24] Z. Wang, Q. Wu, B. Zheng, J. Wang, K. Huang and Y. Shi, "Sequence As Genes: An User Behavior Modeling Framework for Fraud Transaction Detection in E-commerce," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, 2023.

[25] S. Samet, M. T. Ishraque, M. Ghadamyari, K. Kakadiya, Y. Mistry and Y. Nakkabi, "TouchMetric: a machine learning based continuous authentication feature testing mobile application," *International Journal of Information Technology*, vol. 11, no. 4, pp. 625-631, 2019.

[26] S. K. Mandru, "How AI can improve identity verification and access control processes," *Journal of Artificial Intelligence & Cloud Computing*, vol. 1, no. 4, pp. 1-5, 2022.

[27] Q. Yang, G. Peng, D. T. Nguyen, X. Qi, G. Zhou, Z. Sitová, P. Gasti and K. S. Balagani, "A Multimodal Data Set for Evaluating Continuous Authentication Performance in Smartphones," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14)*, Memphis, Tennessee, 2014.

Daniel-Marian DĂNILĂ graduated from the Faculty of Cybernetics, Statistics, and Economic Informatics in 2023, and in 2025 he completed a Master's degree in Information Security at the Bucharest University of Economic Studies. He has been employed by IBM Romania since 2021.

Natalia Sierra is an undergraduate student at Utah Tech University majoring in Information Technology with an emphasis in Cybersecurity. She wrote this paper as part of the SET International Fellowship, a year-long research program supporting undergraduate scholarship. Natalia has been named to the President's List twice in recognition of academic achievement and has a strong academic interest in cyber defense strategies.

Alin ZAMFIROIU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2009. In 2011 he has graduated the Economic Informatics Master program organized by the Bucharest University of Economic Studies and in 2014 he finished his PhD research in Economic Informatics at the Bucharest University of Economic Studies. Currently he works like a Senior Researcher at "National Institute for Research & Development in Informatics, Bucharest" and associate professor at the Department of Economic Informatics and Cybernetics at the Bucharest University of Economic Studies, Bucharest. He has published as an author and co-author of journal articles and scientific presentations at conferences.