Adaptive Continuous Authentication System for Smartphones using Hyper Negative Selection and Random Forest Algorithms

Maryam M. Alharbi, Rashiq Rafiq Marie
College of Computer Science and Engineering
Taibah University, Medina, Saudi Arabia

***Abstract*—As smartphones have become a part of our daily lives, including payment and banking transactions; therefore, increasing current data and privacy protection models is essential. A continuous authentication model aims to track the smartphone user's interaction after the initial login. However, current continuous authentication models are limited due to dynamic changes in smartphone user behavior. This paper aims to enhance smartphone user privacy and security using continuous authentication based on touch dynamics by proposing a framework for smartphone devices based on user touch behavior to provide a more accurate and adaptive learning model. We adopt a hybrid model based on the Hyper Negative Selection Algorithm (HNSA) as an artificial immune system (AIS) and the random forest ensemble classifier to instantly classify a user behavior. With the new approach, a decision model could detect normal/abnormal user behavior and update a user profile continuously while using his/her smartphone. The proposed approach was compared with the v-detector and HNSA, where it shows a high average accuracy of 98.5%, a low false alarm rate, and an increased detection rate. The new model is significant as it could be integrated with a smartphone to increase user privacy instantly. It is concluded that the proposed approach is efficient and valuable for smartphone users to increase their privacy while dynamic user behaviors evolve to change.**

***Keywords*—Continuous authentication (CA); artificial immunes system (AIS); negative selection algorithm (NSA); random forest algorithm (RFA); smartphones**

I. Introduction

According to the Global System for Mobile Communications Association (GSMA) Intelligence Reports, there are more than 5 billion smartphone users in the world today [1]. Smartphones are the most sensitive and essential device in all aspects of our life, including memories, sensitive data, buying, education, and work. Consequently, there is a high need to increase the privacy and security in smartphones by authentication models. The traditional authentication methods in smartphones depend on PIN authentication approaches, known as entry-point authentication, as they authenticate the user only at the beginning of a particular session. Generally, the entry-point authentication models include PIN authentications, passwords, or biometrics; however, these models are considered discontinuous because they do not track smartphone access while interacting with the system.

Therefore, smartphones could have adversarial attacks after the initial authentication if such authentication models are adopted. On the other hand, continuous authentication keeps a consistent track of a smartphone's access over time. Continuous authentication has attracted researchers' attention in recent years to secure computers, the Internet of Things (IoT), and mobile phones, where they are commonly used to solve a device authentication problem after user login. Therefore, methods that use extensive logging in features such as behavioral biometrics (such as touch dynamics, keystrokes, movement, walking, and daily activity) are critical to continuous authentication [2].

Despite the ever-increasing number of continuous authentication approaches, there is still a need to develop new techniques applicable in real life where a device owner does not operate with the smartphone continuously. However, there are many open issues in the current continuous authentication studies, including the lack of real-world dataset, the need for increased accuracy, low usability of current models, and lack of adaption of proposed models [3].

One of the prevalent biometric behaviors of smartphone users is touch dynamics, a promising approach recommended by many researchers because it requires no additional hardware to collect information; it has high usability and robust security [3-6]. While biometrics are easy to observe, once breached, they cannot be modified or revoked. It is necessary to lift fingerprints from smooth surfaces (such as smartphone screens or coffee cups). However, the increased proliferation of high-resolution cameras raises the threat of imaging from a distance.

According to the literature, hackers gained access to the German ministers of defense Ursula von der Leyen, using just a couple of high-definition pictures. Therefore, this case undermines the finger's protection since fake fingers may be produced from other materials [7]. Therefore, biometric behavior authentication has several challenges. Mahfouz et al. [2] address the challenges, capabilities, and restrictions associated with biometric behavior authentication. One of the main reported challenges was the intra-class variations between individuals due to the user changing behavior over time; therefore, the user data in the enrollment phase (active use) may vary from the recognition phase (classification). Consequently, a proper continuous system must adhere to the challenges to prevent adversary attacks.

This paper aims to develop a model for continuous authentication (CA) using Hyper Negative Selection Algorithm (HNSA) and the random forest ensemble classifier (boosted) to authenticate and detect illegal smartphone users. This model will be enhanced to be self-adaptive to changes in user behaviors. The proposed approach starts with an ensemble learning model that enhances the performance of HNSA by applying the random forest algorithm in the testing phase (of the HNSA) to maximize the accuracy and improve the prediction to decide the user data type in the model as normal, abnormal, new normal, or new abnormal. Therefore, the proposed Random Forest Negative Selection Algorithm (RFNSA) can adapt by updating continuous data as users interact with the smartphone.

This paper is organized as follows: presents a background of smartphone authentication and negative selection algorithms in Section 2. The Foundation of the artificial immune system shows in Section 3. Section 4 presents the related works to this study. The methodology of this work is discussed in Section 5. The results obtained are illustrated in Section 6. The results are discussed in Section 7. The last section concludes the paper and presents limitations and future work.

II. Background and Related Work

A. Smartphone Authentication System

This study classifies authentication systems into continuous or discontinuous models based on how they track user interaction during smartphone usage.

1) Discontinuous authentication systems: The mobile authentication process could be categorized into three different categories: knowledge-based, possession-based, and identity-based authentication models [8]. Knowledge-based models depend on patterns that a mobile owner knows, such as passwords (such as a PIN or passcode). A possession-based model is based on an attribute that the owner has, such as a key to a lock or One-Time Password (OTP) [9], while identity-based authentication is based on something that identifies the mobile owner. Literature refers to combinations of authentication methods to a smartphone as authentication factors (AF).

Regardless of which category is used, the passcode of possession-based models is considered an entry-point authentication method as it does not follow the user's actions while the user is not actively using the smartphone. Passwords in smartphones could be in three forms: textual, digital, or graphical. On the first hand, a textual password consists of a single or intentional mixture of letters (A-Z), digits (0 to 9), symbols. A PIN (or a digit passcode) consists of numerical symbols with a typically 4–6-digit PIN as a common authentication factor. On the other hand, the graphical passcode consists of either a drawing (e.g., DooDB) or a linked dot series on a virtual grid interface [10]. The different authentication techniques could be used in smartphones, namely, slide lock, number lock, graphical-based passwords, fingerprint, and face recognition authentications [8].

However, these methods suffer from being attacked if the device is kept isolated; therefore, these discontinuous authentication methods are insecure.

2) Continuous authentication systems: Continuous authentication, also called transparent, active, or implicit authentication, is an implicit way of verifying the authenticated user using mobile system features and built-in mobile sensors that track their users' behavioral attributes [11]. The intuition behind the behavioral approach is based on the distinctive user patterns commonly used in an authentication task. While users interact with their smartphones, the device implicitly captures their interaction with the device, including user touch patterns, environmental and sensory data [12], [13]. The collected user's behavioral data (biometrics) works without knowing or explicitly asking to enter specific data. The goal is to improve mobile security continuously and transparently throughout the entire routine session [13].

Continuous authentication differs from entry-point (or traditional) authentication by two main characteristics: continuity and transparency [2]. The continuity verifies that a user is legitimate as long as the user uses the smartphone; therefore, it is an automatic re-authentication process. The second property, transparency, allows the authentications to be executed seemingly without interrupting the user. Since a CA system continually tests logged-in users' identity, it is more reliable, easy to use, stable, and encourages schemes with several protection layers. CA methods allow multi-layers of authentication of smartphones to log in to the device down to a specific application based on preference. Researchers see a somewhat favorable reaction to multiple-level authentication schemes for mobile applications that can be a welcome addition to existing conventional mobile operating systems [14].

B. Negative Selection Algorithm

The Negative Selection (NS) is an immune system mechanism that prevents self-reactive lymphocytes from being formed. As a result, only those lymphocytes that do not strongly bind with self-antigens survive this selection process. The NS theory motivated Forrest et al. [15] to suggest a generic negative selection algorithm for detecting data anomalies. It was further extended for detecting network intrusions and using it widely in computer security and fault detection. The basic concept is to apply a collection of detectors in the corresponding space and classify the data as self or non-self. To apply NS's mechanism, the shape space U is subdivided into (S), denoted as self, and (N) for non-self.

$$U=S\cup N \text{ and } S\cap N=\emptyset. \quad (1)$$

The negative selection algorithm divided into two main stages, as shown in Fig. 1, namely, Generating detectors stage and the Detection stage. The aim of the first stage is censoring to generate the valid detectors' set by generating the random detectors. In this stage, if the random detector matches the self then eliminated. Random detectors that do not match any self-data will be store in a detector set to use in the second stage. In the second stage, called (Detection stage) the monitor of protected self-data by comparing the self-set (S) with the

Detectors set (D). If the detector matches with data, which means the data is classified as non-self.

Many families of the negative selection algorithm have been developed, which keeps the main characteristics of the first version of the negative selection algorithm proposed by [15]. However, the main drawbacks of the first version of the negative selection algorithm (classical NSA) are the time-consuming and complexity of space. The NSA is divided into two types based on data representation: binary negative selection algorithm (BNSA) and real value negative selection Algorithm (RNSA). The newest version of the negative selection algorithm proposed by Zhou and Dasgupta [16] is v-detector which is now widely used as a framework for many studies due to its advantages over the previous versions with constant size detectors. The detectors' size in the v-detector varies from one detector to another.

Fig. 2 shows the difference between constant and variable size detectors in 2D space. The grey color in the figure represents the region of self, which is using as a self-sample for data training data. The circles represent the detectors covering the region of non-self while the black holes present the non-covering area; using variable-sized detectors, the greater region of non-self-area can be filled with fewer detectors, whereas smaller detectors can fill the gaps [16].

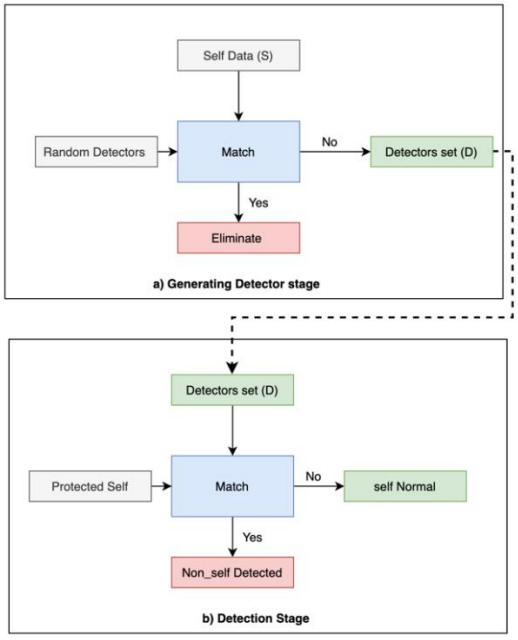

Fig. 1. The Main Stages of NSA Implementation.

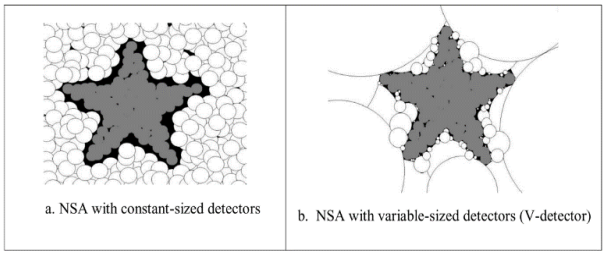

Fig. 2. The difference between Constant Size and Variable Size Detectors [16].

Recently, many NSA enhancements have been developed to address the shortcomings of the classical version, where most of them are based on how detectors are regenerated. However, not only the detector size but also its coverage is hardly achieved. Moreover, recent improvements were proposed to improve the NSA's efficiency, expand its scope, and overcome its limitations [17].

Ramdane [18] proposed an NSA-based hybrid adaptive mechanism for computer network intrusion detection called HNSA–IDSA (hybrid NSA). The proposed approach has a high ability to adaptive learning when changes happen in the system profile. The uniqueness of HNSA is that normal and abnormal self-detectors are created at the training stage by using both normal and abnormal data. In the proposed NSA's test process, its status about the *normal* and *abnormal* profile is defined by the studied sample class. The tested sample in that model's testing phase is based on its position in the system profile (normal and abnormal). Therefore, the HNSA study mechanism seems very useful to adapt system profiles when a different normal or abnormal activity is observed.

There are limited studies that apply artificial immune systems to continuous authentication in smartphones. The work of [19] demonstrated that NSA is an approach that is very appropriate for continuous authentication for PCs. The negative selection method can regularly track any changes in the environment, depending on the computer system owner's interaction. The researchers included both mouse, keystroke users' biometric activities and evaluated the suggested NS's precision; the highest reported accuracy was 99.7%. To the researchers' knowledge, the first continuous authentication method that is based on the AIS class of algorithms was by [20]. They proposed a CA method based on AIS using the Clonal Selection (CS) algorithm for smartphones. The suggested approach was extended to a dataset of screen touch patterns on smartphones; they performed the CS experiment and got 93.81% average accuracy.

III. Methodology

This paper builds a model for continuous authentication (CA) by applying the NSA on the dataset of [21], [22]. To make the model in high adaptability, the proposed model, Random Forest Negative Selection Algorithm (RFNSA), is an enhanced version of the Hyper Negative Selection Algorithm (HNSA) by applying the random forest algorithm to maximize the accuracy and improve the prediction to decide the data type in the model as normal, abnormal, new normal, or new abnormal. The proposed approach (RFNSA) can adapt by updating continuous data as users interact with the smartphone. As shown in Fig. 3, the research framework has four phases: Data preparation, Training, Testing, and Decision.

A. Data Preparing Phase

The proposed model (RFNSA) requires both positive (*normal*) and negative (*abnormal*) data samples to generate the user profile. Every user has their behavior, which identifies their identity.

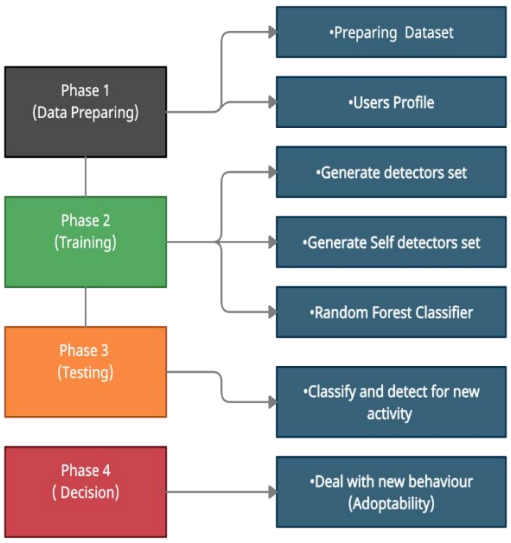

Fig. 3. Research Methodology Procedures.

The user profile contains normal behavior and abnormal behavior. Due to the advantage of negative selection algorithm (NSA), it only needs positive data to detect the identity of the user. The abnormal behaviour generated using v-detector algorithm. This paper focuses only on the first eight features of the adopted dataset (pressure, size, touch major, touch minor, duration, fly time, shake, and orientation) as shown in Table I.

For example, the area covered by the user touch (size) and the time duration (pressure) could detect user behavior. These behavioral characteristics help to understand the user's interaction with the smartphone. Because the features have different ranges value (scales), there is a possibility that higher weighting would be assigned to features with higher magnitude, which might affect the performance of the algorithm. Therefore, values were normalized to be in the range of (0 and 1), often defined as minimum/maximum scaling. A new column was added to the dataset, a label of 0 for self and 1 for non-self. Once data is prepared, the dataset was split into 70% for training and the rest of the data for testing and evaluating model performance.

B. Data Training Phase

1) Initial training: After the dataset preparation and generating user profile, the second research procedure is training phase that is divided into two sub-phases: initial training (to cover the non-self-area with non-self-detectors) and further training (to cover the self-area with self-detectors). The role of the initial training is to generate the non-self-detectors set, whereas generating the self-detectors set is executed in further training sub-phases.

The purpose of this procedure is generating the non-self-detectors set. The v-detector is an enhanced version of the NSA with control parameters: the self-radius, estimated covering, and maximum number of detectors [16]. In this algorithm, the random detectors are generated—the detectors produced by randomly generated numbers (candidates' detectors) one by one. The Euclidean distance between the random detectors and a self-point is calculated to recognize its relation. The shortest distance indicates high affinity, which means the detectors match the self-point. The detector is eliminated, and a new random detector is created if the distance to the nearest self-point is less than the value of the self-radiusr_s. If the minimum distance between the random detector and self-point is more than the self-radiusr_s, the detector is temporarily stored. The radius of this detector is recorded as r_d with the value of minimum distance to the nearest self-point. As shown in Fig. 4, represent how the detectors in shape space with variable size to cover non-self-area region.

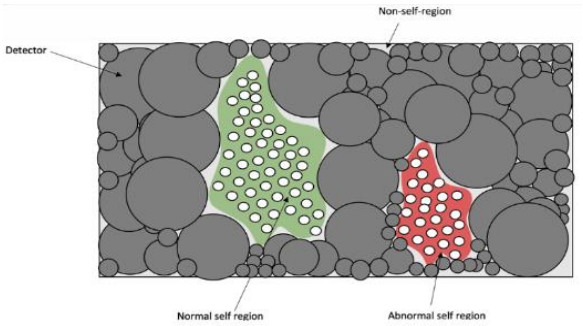

Fig. 4. Normal and Abnormal Regions.

TABLE I. A SAMPLE OF USERS' DATASET

#	Pressure	Size	Touchmajor	Touchminor	Duration	Flytime	Shake	Orientation
0	0.738576	0.433604	9.063023	3.287221	300.8392	145.5518	907.933743	1
1	0.490881	0.493693	4.220438	9.336123	635.8852	484.2957	588.150531	1
2	0.810864	0.493129	6.970505	9.831867	188.5718	464.8582	963.755181	1
3	0.474932	0.355566	2.63251	2.806116	979.9968	638.5064	245.999725	1
4	0.100484	0.184131	2.32429	6.395209	436.6558	653.1836	410.864218	1
...	...	...	...	...	...	...	...	...
96171	0.093137	0.084314	14.32297	14.32297	76	388	47.40989	1
96172	0.118954	0.045752	7.772156	7.772156	64	0	28.167099	1
96173	0.103268	0.047059	7.994217	7.994217	67	579	52.578495	1
96174	0.084314	0.036275	6.16221	6.16221	98	1262	30.075933	1

The generating detectors process in v-detector is controlled by three parameters: radius of self-point (r_s), max number of detectors (D_{max}), and the estimated coverage (M_{max}). The last two parameters are the termination criteria. The D_{max}, are determined to allow the maximum allowed detectors that could be generated. The generated detectors are temporarily stored to check if it matches any previously stored detectors. If it is found in already stored detectors, then it eliminated, and the maximum coverage area M_{max} is incremented. If it is not stored previously, it added to the detectors set, and each time it is permanently stored detector increment the parameter D_{max} and reset the counter M_{max} to zero. If the counter of consecutive attempts that fell on protected points exceeds a limit M_{max}, the generation stage concludes with enough assurance that the coverage is adequate to protect the non-self-area [23].

2) Further training: Further training was applied to generate the self-detectors set (SD). The benefit of adopting this operation is to reduce the next phase's computational cost (testing phase). The further phase works as follows:

- Use ***D*** as a training set to generate self-detectors (SD) for the self-area; the self-detector is defined as shown in equation 2, where SND is the set of self-normal detectors, SAD is the set of self-abnormal detectors.

$$SD = SND \cup SA \tag{2}$$

- Generate random values for self-detectors; if self-detectors SD match any detectors D as shown in equation 3 wherer_d, the radius of the detector is; then, the detectors are eliminated.

$$dis(SD, D) < r_d \tag{3}$$

- If the random self-detector does not match any detector, calculate the Euclidian distance for all detectors in the training set (D) and find the minimum distances using equations 4,5 where r_{sd} is the radius of self-detector:

$$dis(SD - D) = \sqrt{\sum_{i=0}^{n}(SD - \mathrm{D})^{\,2}} \tag{4}$$

$$r_{sd} = MinDis(SD, D) \tag{5}$$

- Add a detector to the self-detectors' set, SD, with a label that indicates the type of detector. If the detector is situated in a normal region, it is labeled normal (0); otherwise, it is labeled abnormal (1).
- In this stage, all self-sample and generated new self-detectors will be considered self-detectors with two types (normal self-detectors and abnormal self-detectors).

C. Testing Phase

Random Forest Algorithm (RFA) is a supervised learning algorithm that is a more precise forecast. The basic concept of RFA is to ensembles many decision tresses to vote the best prediction. In the proposed enhanced model, training is carried on the training data to enhance the model performance to predict the data type (normal or abnormal). The random forest has many advantages that make it an excellent choice to apply it to the proposed model, such as high accuracy, quick prediction, fast training, and the ability to handle unbalanced data to minimize the error rate. The work of [18] applies this procedure to classify the income data and detect new behavior in IDS.

This research follows up the same steps with engaging the random forest classifier to enhance the preprocessed data classification, in this phase, the process using self-detectors (SD) which contains self-normal detectors (SND) and self-abnormal detectors (SAD) with test sample set (T) following equation 6.

$$SD = SND \cup SAD \tag{6}$$

To classify the test sample data type, the distance between the test sample points (t) to self-detectors (SD) is calculated as shown in equation 7, which is used to decide the data type as known abnormal, known normal, new abnormal, or new normal.

$$dis(t - SAD) = \sqrt{\sum_{i=0}^{n}(\mathrm{t} - \mathrm{SAD})^{\,2}} \tag{7}$$

$$mindis(t - SAD)/r_{asd} \leq 0.9 \tag{8}$$

If $\mathrm{dis}(\mathrm{t} - \mathrm{SAD}) < \mathrm{r_{sad}}$ where $\mathrm{r_{sad}}$ is the radius of abnormal self-detector, then the prediction is checked from a random forest classifier; if it is the same (abnormal), then an additional condition is added (equation 8) to make sure the sample t in the abnormal region. If all conditions are met, the data type is known abnormal; otherwise, the data type is one of the remaining three types.

$$dis(t - SND) = \sqrt{\sum_{i=0}^{n}(\mathrm{t} - \mathrm{SND})^{\,2}} \tag{9}$$

$$mindis(t - SND)/r_{nsd} \leq 0.9 \tag{10}$$

If $\mathrm{dis}(\mathrm{t} - \mathrm{SND}) < \mathrm{r_{nsd}}$ where $\mathrm{r_{nsd}}$ is the radius of the normal self-detector, then the prediction is checked from a random forest classifier; if it is the same (normal), then an additional condition is added (equation 10) to make sure the sample t in the normal region. If all conditions are met, the data type is known normal; if there is no match, the data type will be one of the remaining two data types.

If any self-detectors do not cover t, the model classifies it as new normal or new abnormal based on the nearer region to this point t.

$$if(Mindis(t, SAD) - r_{asd} \leq Mindis(t, SND) - r_{nsd} \tag{11}$$

If the condition in equation 11 met, then the t is considered abnormal because it is nearer to the region of abnormal more than the normal region.

$$if(Mindis(t, SAD) - r_{nsd} \leq r_d \tag{12}$$

where $\mathrm{r_d}$ is the radius of detector, then new abnormal is added to SD, and it is labeled as abnormal with the radius size detector $\mathrm{r_d}$ as a radius of new abnormal; otherwise, new abnormal label is added to SD with the radius of nearest self-abnormal detector $\mathrm{r_{sad}}$ as a radius of new abnormal.

$$if(Mindis(t, SND) - r_{snd} \leq Mindis(t, SAD) - r_{asd} \tag{13}$$

A new normal behavior is detected if equation 13 met, then t is considered normal because it is the nearest to the normal region compared to the abnormal region.

If $(\text{Mindis}(t, \text{SND}) - r_{snd} \geq r_d$ where $r_d$is the radius of the detector, then new normal to is added to the SD, and it is labeled as normal with the radius size of the detector $r_d$as the radius of new normal ; otherwise, new normal label is added to the SD, with the radius of the nearest self-normal detector r_{snd} as the radius of new normal.

To summarize, the proposed algorithm enhances the performance of HNSA by applying the random forest algorithm in the testing phase to maximize the accuracy and improve the prediction. The new approach decides the data type in the model as normal, abnormal, new normal, or new abnormal as follows:

- Train random forest model on self and non-self-data. The random forest will predict the data type (normal or abnormal).
- Predict the test data type to either (normal or abnormal).
- Add a condition to the model to calculate the relation between $\text{dis}(t - \text{SAD})$ and r_{sad} ; if the values are not very close, then the first case is applying (known Abnormal). Additional conditions are added to the model to study the relation between $\text{dis}(t - \text{SND})$ and r_{nsd} ; if the values are not very close, then the second case is applying (known normal).

D. Decision Phase

The decision module responds to update the user profile with a new normal or a new abnormal behavior. If the abnormal data is in the user's profile, then immediately the phone is locked, or the user is silently permitted to continue using the phone. If the user behavior has little change, then this behavior is added to the user's profile. The new addition helps the model adapt to user change behaviors repeatedly and detect the new user behavior if the same situation reappears. By this, the adaptability is enhanced aiming to deal with user change over time when the user behavior indicates a new abnormal or new normal behavior.

By updating the data that comes from user interactions with the smartphone, the modified approach, RFNSA, can detect and adapt to changes in the user profile based on new-normal or new-abnormal experiences. The proposed conceptual approach, as shown in the Fig. 5, begins by collecting behavior features of a user touch screen for a specific period used for training. A user profile builds from the previously collected data. After a user profile is established, while the user uses his/her smartphone, the extracted features are processed to detect a user identity continuously.

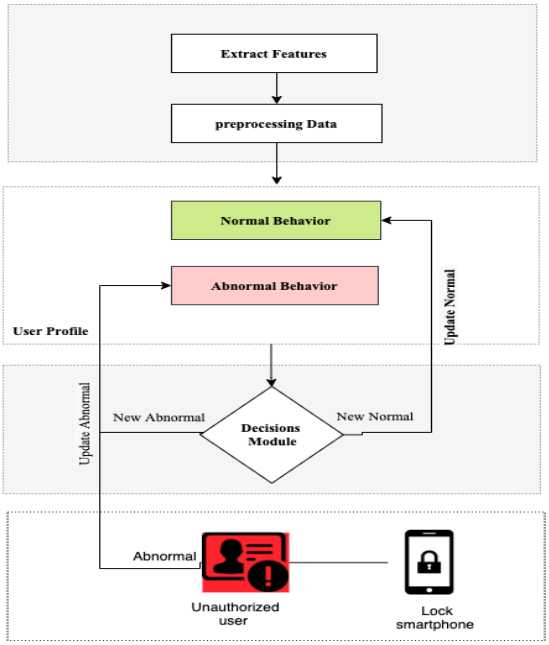

Fig. 5. The Structure Diagram of the Proposed Approach of CA in Smart Phones.

IV. Result

The generated detectors in preparing dataset phase depend on control parameters: radius of self-point (r_s), max number of detectors (D_{max}), and maximum estimated coverage area (M_{max}). The initializing of control parameters affects on generating detector and overall performance. Table II shows the effect of control parameters on proposed algorithm performance.

Due to the high randomness of the v-detector algorithm, many tests were performed for each of the experimental tests; thereby, the experiment took the average value of the number of detectors generated (D (mean)), execution time, and accuracy. Many tests are conducted to choose the optimal value of (D_{max}). Due to the advantages of the nature of v-detector, which is variable-sized detectors to cover the covering area with few detectors, this experiment tests the best value for max number of detectors (from 10 to 22) and noticed that the less number of detectors leads to fast execution without affect the less number of detectors leads to fast execution without effect on the overall of accuracy. The value of D_{max} chosen for this experiment is 16. As described in Table II, a smaller r_s increases the execution time and increases the number of generating detectors.

This experiment uses 90% of the estimated area by using the value of M_{max}=10 in order to make balance on accuracy and less execution time. The non-covering area is $1/m$ = 0.1 from the whole non-self-region, while the maximum number of detectors, D_{max} was 16. The non-self-detectors were generated multiple times to make adequate negative data of the negative sample of users' behavior profile. Approximately 500 records were generated for each user. That is the data building from non-self-detectors not matching the positive data (*normal* behavior).

TABLE II. EFFECTS OF CONTROL PARAMETERS ON GENERATING DETECTORS

r_s	M_{max}	D_{max}	D (mean)	Execution time (s) (mean)	Accuracy
0.1	10	16	16	33.69	99.20
0.2	10	16	17	33.53	99.74
0.3	10	16	16	30.13	99.13
0.4	10	16	11	49.45	99.10
0.01	10	16	18	94.75	96.20
0.02	10	16	15	83.12	97.30
0.03	10	16	12	79.67	96.90
0.04	10	16	17	65.19	99.10

The proposed RFNSA uses *normal* and *abnormal* data for users as (self-set). In this experiment, the dataset split into 70% for training and 30% for testing. The RFNSA model starts with generating the non-self-detectors using the v-detector algorithm described previously. The non-self-detectors set with their radius are used in the next stage to generate the self-detectors. The non-self-detectors were generated using control parameters (r_s=0.2, D_{max}=16 and M_{max}=10), as they were the best performing in terms of average accuracy and execution time.

Similarly, the self-detectors were generated in the same manner using the v-detector; the generated detectors have to match with self-point and not to match detectors from previous procedures. The goal of this stage was to cover the self-area with detectors. At the end of further training, all training data with the generated self-detectors were considered self-detectors, which have two types: normal self-detector (SND), and abnormal self-detectors (SAD). The proposed RFNSA adds additional training to the HNSA with random forest to enhance the model prediction to decide the user's behavior as normal (label 0) or abnormal (label 1). This study used the sklearn library implementation of the random forest classifier on the training dataset. The random forest classifier performance is shown in Fig. 6, which gets a maximum average accuracy of 99.9%. The random forest algorithm is used in the testing phase later to predict the type of data.

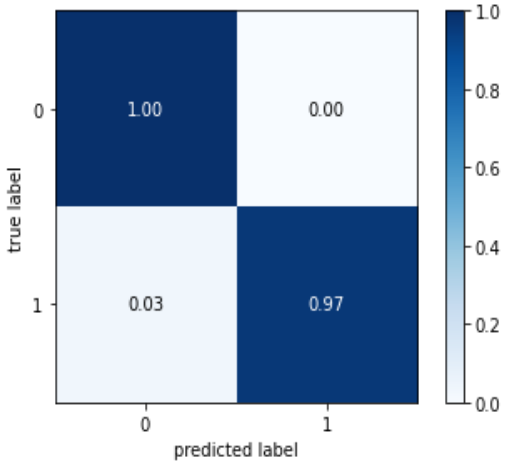

Fig. 6. Confusion Matrix for Random Forest Classifier.

The testing stage classifies the test data into normal or abnormal and detects new user behaviors. As shown in Fig. 7, for the 8th user and out of 2,102 records, the proposed algorithm detects 144 records as new abnormal behavior and 1,941 records as new normal behavior. All those new behaviors were added to the dataset to be recognized next time, which means applying adaptability.

Table III represent the (normal and abnormal) data for each user. As noticed from Table IV**,** the model success in detecting user behavior. The highest error ratio in the classification of test data in user 11 and user 20, where user 11 incorrectly identify about 29 new normal records as new abnormal with error ratio (1.378%), and user 20 identify 6 records from abnormal test data as new normal with error ratio (0.285%).

An efficient authentication system should have a high detection rate (DR) and low false alarm rate (FAR). This paper uses accuracy and detection rate together with false alarm rate to evaluate the model's performance. Due to the unbalanced dataset in two classes (normal and abnormal), the balanced accuracy was used by taking the recall for each class of data.

The performance of the proposed model is shown in Table V, which describes the metrics values for each user, and the average values for all users. The average values of performance metrics appeared to be more acceptable based on the findings, compared with the literature's results of continuous authentication for smartphones.

The experiment results were conducted to compare among three models (v-detector, HNSA, RFNSA). The experimentation was conducted with different self-radius values as shown in Fig. 8; the RFNSA is more stable than HNSA. The HNSA minimum accuracy was 51% when the r_s was 0.4, while the maximum accuracy was (97.7%) when the radius was 0.3.

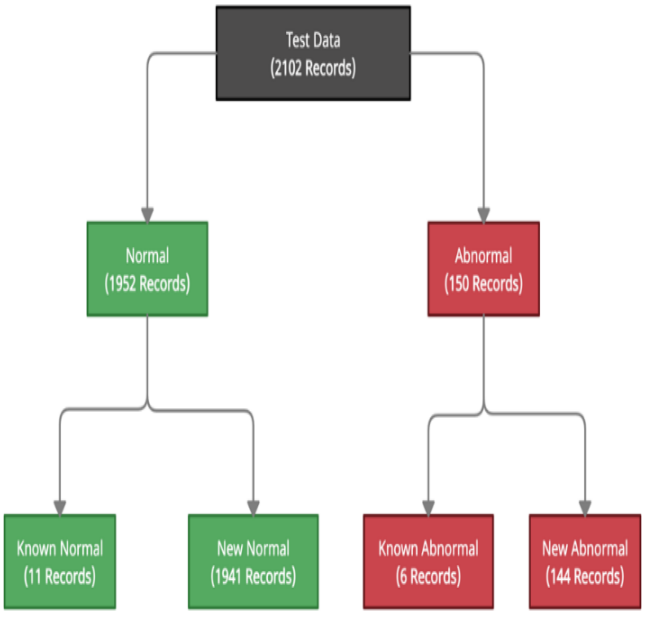

Fig. 7. Example of Classifying Test Data for user 8.

TABLE III. BUILDING USERS' PROFILES

Users	Normal Data	Abnormal Data*
1	12,095	528
2	6,506	507
3	6,506	505
4	6,506	501
5	6,506	506
6	6,506	508
7	6,635	502
8	6,506	500
9	6,506	504
10	6,506	507
11	6,506	503
12	7,196	507
13	6,506	506
14	6,506	503
15	6,506	504
16	6,506	504
17	6,635	502
18	6,506	505
19	6,506	507
20	6,506	507
Total	**136,657**	**10,116**

TABLE IV. CLASSIFYING DATA FOR ALL USERS

User	Normal Data			Abnormal Data			Classifying Error
	Test (Total)	known	New	Test (Total)	Known	New	
1	3,629	0	3634	159	0	154	0.13%
2	1,952	0	1,954	153	0	151	0.09%
3	1,952	8	1,946	151	1	148	0.09%
4	1,952	185	1,768	151	0	150	0.04%
5	1,952	19	1,935	152	1	149	0.09%
6	1,952	16	1,936	153	0	153	0.00%
7	1,952	0	1,952	152	1	149	0.09%
8	1,952	11	1,941	150	6	144	0.00%
9	1,952	0	1,951	150	1	150	0.05%
10	1,952	1	1,954	153	0	150	0.14%
11*	1,952	89	1,883	153	2	183	1.37%
12	1,952	0	1,954	153	0	151	0.09%
13	1,952	0	1,954	151	0	149	0.09%
14	1,952	0	1,954	153	0	151	0.09%
15	1,952	2	1,950	150	0	150	0.00%
16	1,952	92	1,861	150	0	150	0.05%
17	1,952	0	1952	152	0	152	0.09%
18	1,952	0	1954	151	0	149	0.09%
19	1,952	19	1935	152	0	150	0.09%
20*	1,952	35	1923	153	2	145	0.28%

TABLE V. EVALUATION METRICS FOR ALL USERS

User	Acc. (%)	Presc. (%)	Recall (%)	f1-score (%)	(DR) (%)	FAR	Balanced Acc. (%)
1	99.8	100	97	99	97.4	0.0000	98.70
2	99.7	99	99	99	97.38	0.0010	98.64
3	99.6	99	98	99	96.69	0.0015	98.26
4	99.7	99	98	99	98.01	0.0010	98.90
5	98.6	99	98	99	96.72	0.0122	97.70
6	99.1	95	99	97	98.00	0.0076	98.96
7	98.3	99	98	99	96.72	0.0122	97.70
8	98.2	99	99	99	99.00	0.0005	99.70
9	99.8	99	100	99	99.33	0.0015	98.30
10	99.7	100	99	99	97.30	0.0980	99.20
11	98.7	99	97	98	95.40	0.0204	96.68
12	99.1	100	100	100	97.30	0.0000	98.00
13	99.9	100	100	100	98.60	0.0000	98.60
14	99.7	98	100	99	99.00	0.0025	99.80
15	99.6	98	99	99	98.10	0.0025	98.90
16	99.8	99	100	99	98.30	0.0025	99.80
17	99.9	100	100	100	99.00	0.0000	99.70
18	98.2	99	99	99	98.43	0.0122	97.70
19	99.6	100	98	99	95.40	0.0000	97.70
20	99.7	100	98	99	96.07	0.0000	98.03
Avg	**99.33**	**99.05**	**98.8**	**98.9**	**97.60**	**0.0043**	**98.55**

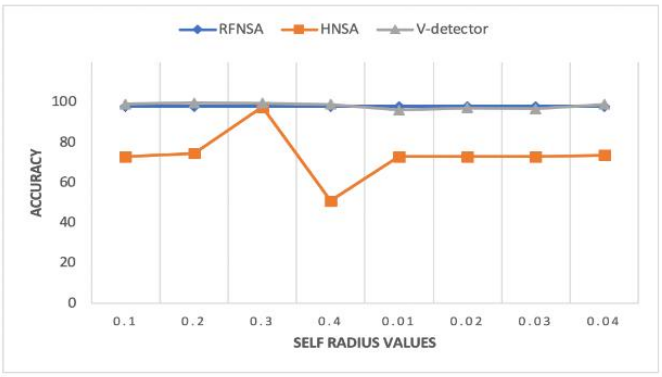

Fig. 8. Performance of RFNSA, HNSA and V-detector (Average for all users).

V. DISCUSSION

Any efficient security system should have the ability to detection for anomaly with low error rates quickly. The proposed model RFNSA would be adopted in security fields in continuous authentication systems in smartphones. This study results indicate that this model has a high ability to deal with anomaly user's behavior with self-adaptive ability to deal with changes in users' behavior. However, the current approach is yet to be taken cautiously due to the dataset and its limited features. Moreover, the model has not been applied in practice; therefore, its robustness is not yet tested. Compared with NSA,

HNSA this model has robust and stable performance among various self-radius values. The proposed RFNA stability is due to the effect of the addition of the random forest algorithm and best-chosen values of r_d as $r_d = 2 * r_s$ which was based on the experimental in the HNSA[18]. In the future more values of r_d will be conducted to study the effect of varity r_d on the model performance.

The main critical issue is building a negative sample (abnormal user behavior dataset) to handle these issues; the v-detector can generate the detectors that do not match self-data. Those detectors were considered abnormal data and were generated for each user. The proposed model deals with new behavior by adding this behavior to the user profile, but this process should, in contract, deal with delete any old behavior, which is matching user behavior for a long time. The deletion of user profile data was kept for future research. Tuning the proposed algorithm to the best performance was time-consuming; therefore, based on previous research, specific values were chosen; the value for M_{max}=10 to cover the whole non-self-region (90%) in a short time and D_{max}=16 for the advantages of v-detectors that generate variable-sized detectors to cover the estimated coverage area quickly.

VI. Conclusion, Limitations and Future Work

A. Conclusion

In this paper, a continuous authentication framework proposed depends on touch dynamic, where users interact with their device's touchscreen. This study applies a modified version of HNSA in an ensemble classifier to detect any user behavior changes while using the smartphone. The HNSA was modified to work better for smartphones over a selected continuous authentication in smartphones' dataset.

The enhanced version is called Random Forest Negative Selection Algorithm (RFNSA). This proposed model gives a stable and efficient performance more than V-detector and HNSA algorithms. One of the framework's most advantages is that it resolves the two biggest challenges in continuous authentication: accuracy and adaptability. The model provides a balanced accuracy of (98.5%), with a high detection rate (97.6 %), low false alarm rate (0.004%), and it adapts itself while a user is using his/her smartphone.

B. Limitations and Future Work

The results obtained from this work give a high accuracy result with a high detection rate and low false alarm rate, but this work still has a limitation that opens the way for several future research possibilities. First, due to the lack of available public datasets, one dataset was used. As a potential future first step, we must comprehensively evaluate our model with more than one dataset to generalize this research's findings. Also, while a new layer was added to increase the authentication model's accuracy, the processing is increased due to the ensemble learner (the random forest). Although the extra processing of random forest might be negligible when a tradeoff is made with security, it must be integrated with the HNSA algorithm. Additionally, combining touch patterns with other behavior measurements can help characterize and verify an improved user behavior.

References

[1] GSM Association, "Mobile economy," GSM, 2020. https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf (accessed Jun. 01, 2021).

[2] A. Mahfouz, T. M. M. Mahmoud, and A. S. S. Eldin, "A survey on behavioral biometric authentication on smartphones," Journal of Information Security and Applications, vol. 37, no. October, pp. 28–37, 2017, doi: 10.1016/j.jisa.2017.10.002.

[3] S. Ayeswarya and J. Norman, "A survey on different continuous authentication systems," International Journal of Biometrics, vol. 11, no. 1. pp. 67–99, 2019, doi: 10.1504/IJBM.2019.096574.

[4] G. Dahia, L. Jesus, and M. Pamplona Segundo, "Continuous authentication using biometrics: An advanced review," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 10, no. 4. pp. 1–23, 2020, doi: 10.1002/widm.1365.

[5] 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.

[6] M. A. Alqarni, S. H. Chauhdary, M. N. Malik, M. Ehatisham-ul-Haq, and M. A. Azam, "Identifying smartphone users based on how they interact with their phones," Human-centric Computing and Information Sciences, vol. 10, no. 1, p. 7, 2020, doi: 10.1186/s13673-020-0212-7.

[7] G. Paul and J. Irvine, "Fingerprint Authentication is here, but are we ready for what it brings?," IEEE Consumer Electronics Magazine, vol. 5, pp. 79–83, 2016.

[8] U. Shafique et al., "Modern Authentication Techniques in Smart Phones: Security and Usability Perspective," International Journal of Advanced Computer Science and Applications, vol. 8, no. 1, pp. 331–340, 2017, doi: 10.14569/ijacsa.2017.080142.

[9] P. S. Teh, "Using Users' Touch Dynamics Biometrics to Enhance Authentication on Mobile Devices," The University of Manchester (United Kingdom), 2019.

[10] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "The DooDB Graphical Password Database: Data Analysis and Benchmark Results," IEEE Access, vol. 1, pp. 596–605, 2013, doi: 10.1109/ACCESS.2013.2281773.

[11] J. M. Jorquera Valero et al., "Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System," Sensors, vol. 18, no. 11. 2018, doi: 10.3390/s18113769.

[12] A. I. Filippov, A. V Iuzbashev, and A. S. Kurnev, "User authentication via touch pattern recognition based on isolation forest," in 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018, pp. 1485–1489, doi: 10.1109/EIConRus.2018.8317378.

[13] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics : A Contemporary Survey," pp. 1–19, 2020.

[14] S. Rasnayaka and T. Sim, "Who wants Continuous Authentication on Mobile Devices?," in 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2018, pp. 1–9, doi: 10.1109/BTAS.2018.8698599.

[15] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Self-nonself discrimination in a computer," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pp. 202–212, 1994, doi: 10.1109/risp.1994.296580.

[16] Z. Ji and D. Dasgupta, "V-detector: An efficient negative selection algorithm with 'probably adequate' detector coverage," Information Sciences, vol. 179, no. 10, pp. 1390–1406, 2009, doi: 10.1016/j.ins.2008.12.015.

[17] C. Ramdane and S. Chikhi, "Negative Selection Algorithm : Recent Improvements and Its Application in Intrusion Detection System," vol. 6, no. 2, pp. 20–30, 2017.

[18] C. Ramdane, "A new negative selection algorithm for adaptive network intrusion detection system," International Journal of Information Security and Privacy, vol. 8, no. 4, pp. 1–25, 2014, doi: 10.4018/IJISP.2014100101.

[19] O. Aljohani, N. Aljohani, P. Bours, and F. Alsolami, "Continuous Authentication on PCs using Artificial Immune System," in 2018 1st International Conference on Computer Applications Information Security (ICCAIS), 2018, pp. 1–6, doi: 10.1109/CAIS.2018.8442022.

[20] N. Aljohani, J. Shelton, and K. Roy, "Continuous Authentication on Smartphones Using An Artificial Immune System," pp. 171–174, 2017.

[21] N. Shekoufa, J. Rahimipour Anaraki, and S. Samet, "Replication Data for: Continuous Authentication using Touch Dynamics and its Application in Personal Health Records." Harvard Dataverse, 2019, doi: doi/10.7910/DVN/VVXWZO.

[22] N. Shekoufa, "Continuous authentication and its application in personal health record systems," Memorial University of Newfoundland, 2017.

[23] S. Dixon, "Studies on Real-valued Negative Selection Algorithms for Self-nonself Discrimination: A Thesis," California Polytechnic State University, 2010.